

M12-03 · GIBALJA VISUAL MANUAL

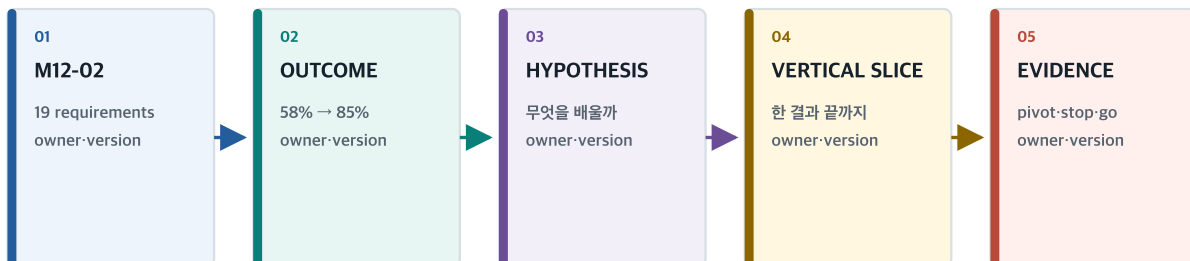
MVP 범위와 우선순위 정하기

한 문장 목표: M12-02 requirement·constraint → product goal·primary hypothesis → whole problem → smallest end-to-end vertical slice + viability floor → evidence priority + dependency + capacity → smallest sufficient test + decision rule → MVP Scope Gate → M12-04 handoff 를 끊김 없이 연결합니다.

M12-03 · 01

MVP는 기능 수가 아니라 학습 사슬로 정의합니다

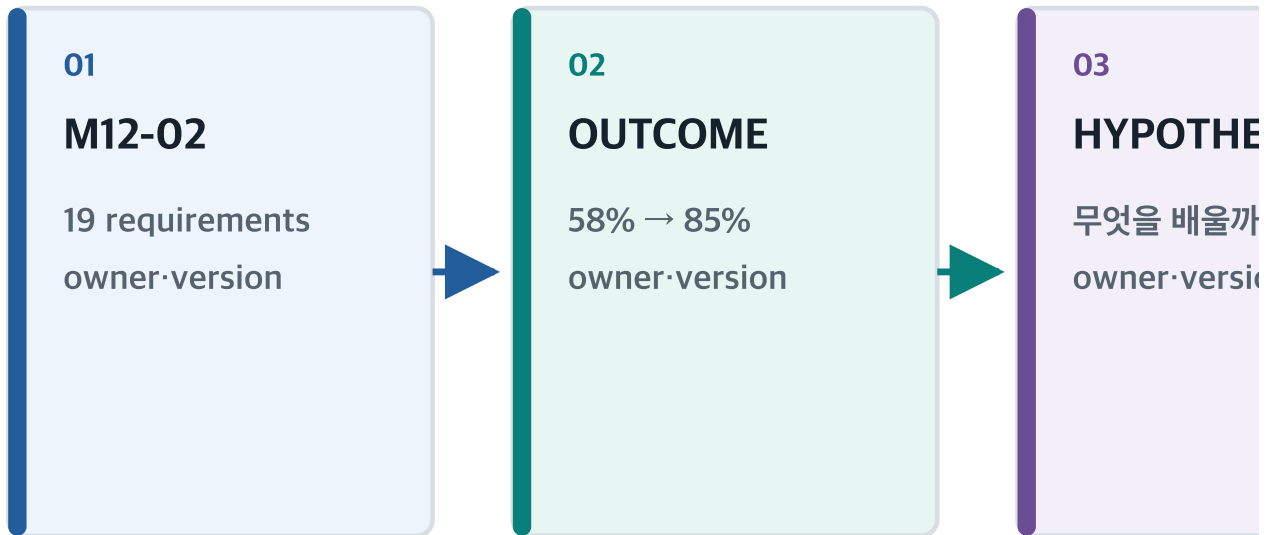
준비된 요구를 결과·가설·완전한 slice·증거·결정으로 연결하고 다시 갱신합니다.



최소 effort의 목적은 가장 중요한 불확실성을 줄여 다음 결정을 바꾸는 것입니다.

그림 1. MVP는 가장 적은 기능 묶음이 아니라 가장 중요한 가설을 최소 effort로 검증하는 viable한 학습 사슬입니다.

준비된 요구를 결과·가설·완전한 slice·증거·결정으로 연결하고 다시 갱신함



니다.

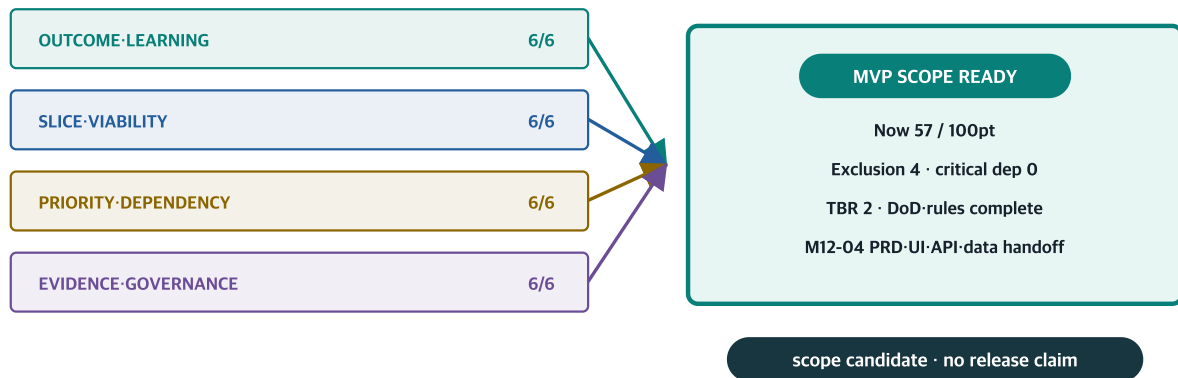


0. 한눈에 보는 MVP Scope Gate

M12-03 · 14

네 lane과 비기능 Gate가 모두 통과해야 범위 후보가 됩니다

24 scenario·12 control, outcome·slice·floor·priority·test 100%, now 57점, 제외 4, critical dependency 0을 함께 판정합니다.



mvp_scope_ready는 M12-04 문서 연결 후보이며 실제 MVP·pilot·release·예산·투자 승인이 아닙니다.

그림 14. 네 lane 6/6, 비기능 coverage 100%, now 57/100점, exclusion 4, critical dependency 0일 때만 scope candidate가 됩니다.

24 scenario·12 control, outcome·slice·floor·priority·test 100%, now 5

OUTCOME·LEARNING	6/6
SLICE·VIABILITY	6/6
PRIORITY·DEPENDENCY	6/6
EVIDENCE·GOVERNANCE	6/6

7점, 제외 4, critical dependency 0을 함께 판정합니다.



MVP SCOPE READY

Now 57 / 100pt

Exclusion 4 · critical dep 0

TBR 2 · DoD·rules complete

M12-04 PRD·UI·API·data handoff

scope candidate · no release claim

학습 순서	시간	남기는 evidence
그림 14장	45분	MVP 오해·가설·slice·priority·Gate 전체 지도
개념·합성 사례	105분	15 scope item·4 exclusion·decision rules
실습 스튜디오	90분	24 scenario·57 audit·6 regression
템플릿·셀프 테스트	60분	내 scope candidate·30문항 답

본문의 사용자·조직·요구사항·scope·수치·비용·시스템은 모두 허구의 합성 학습 자료입니다.

mvp_scope_ready 는 실제 MVP·build·public pilot·production release·예산·투자·계약·보안·접근성 인증 또는 승인을 대신하지 않습니다.

0.1 먼저 기억할 열두 문장

MVP는 최소 기능이 아니라 최대 validated learning을 위한 최소 effort다.
 Minimum보다 viable과 learning을 먼저 묻는다.
 Prototype·PoC·pilot·MVP·release는 서로 다른 계약이다.
 가장 위험한 가설을 먼저 시험한다.
 사용자 문제의 시작과 끝을 끊지 않는다.
 폭은 줄여도 UI·API·data·권한·오류·evidence의 깊이는 잇는다.
 품질 floor는 feature score와 거래하지 않는다.
 포함 범위와 명시적 제외를 함께 쓴다.
 RICE의 숫자보다 입력 source와 range를 보존한다.
 Dependency는 owner·due·sequence·fallback으로 쓴다.
 Metric에는 persevere·pivot·stop rule이 붙는다.
 mvp_scope_ready는 build·pilot·release 승인이 아니다.

1. 이 PDF를 공부하는 방법

1.1 1회차 · 그림과 caption만 읽기 · 45분

그림마다 무엇을 결정하려는가 → 어떤 evidence가 필요한가 → 무엇을 잘못 줄이면 안 되는가 세 지점을 손으로 짚습니다.
 다음 문장을 채웁니다.

이 그림의 계약이 없으면 ____을 MVP라고 오해하고, ____ evidence 없이 ____ 결정을 내린다.

1.2 2회차 · 합성 사례 실습 · 90분

MVP 범위 실습 생성기를 실행합니다.

```
./02_Labs/G12_Product_Definition/L12-03_create-mvp-scope-practice.sh
```

Version	Pass	Now effort	Floor	Critical dependency	Decision
small-feature-cut-v1	6/24	88/100	0%	3	blocked_feature_cut
score-only-priority-v2	15/24	74/100	62%	1	blocked_score_theater
evidence-led-mvp-v3	24/24	57/100	100%	0	mvp_scope_ready

1.3 3회차 · 내 scope candidate 만들기 · 120분

MVP 범위·우선순위·학습 템플릿을 아래 순서로 채웁니다.

1. M12-02 handoff·product goal·outcome
2. Value·usability·feasibility·viability assumption
3. Whole problem·artifact stage·vertical slice
4. Quality floor·scope inventory·exclusion
5. Priority evidence·trade-off·dependency·capacity
6. Test·metric·decision rule·version·M12-04 handoff

1.4 막힐 때

MVP 범위·우선순위 용어집 300에서 지금 그림과 같은 번호의 20개만 읽습니다. 용어마다 **SCP-xx** , **SC-xx** , **CTRL-xx** 중 하나를 붙여야 학습이 실제 계약으로 돌아옵니다.

2. MVP를 다시 정의하기

MVP LEARNING CONTRACT 가장 중요한 제품 가설에 답하기 위해 필요한 최소 effort로, primary user가 whole problem 안의 한 결과를 end-to-end로 달성하고 quality·access·security·privacy·ops floor를 지키며, 결과에 따라 persevere·pivot·stop 결정을 갱신할 수 있는 versioned scope candidate입니다.

2.1 ‘최소’보다 먼저 물을 세 질문

질문	좋은 답	나쁜 축소
무엇을 배울까	Primary hypothesis와 바뀔 decision	기능 수를 줄인다
무엇이 viable한가	Whole outcome·quality floor·recovery	Happy path demo만
어떤 evidence면 충분한가	Metric·threshold·segment·owner	좋은 feedback 몇 개

2.2 합성 case contract

Case: team-follow-up-mvp-scope-2026-07-v1

Primary user: 5~20명 규모 팀의 운영 조정 담당자

Product goal: 팀이 회의에서 결정한 후속 조치를 빠짐없이 책임·기한과 함께 실행한다

Primary hypothesis: 운영 조정 담당자가 한 회의 source에서 실행 기록을 만들고 제안된 담당자가 담당·기한을 확인하면 완전 기록률이 높아지고 재입력 시간이 줄어든다.

Outcome: complete 58%→85% · re-entry 14→5 min

Guardrail: misassignment ≤2% · no group worse · sensitive incident 0

Boundary: synthetic future limited learning pilot candidate

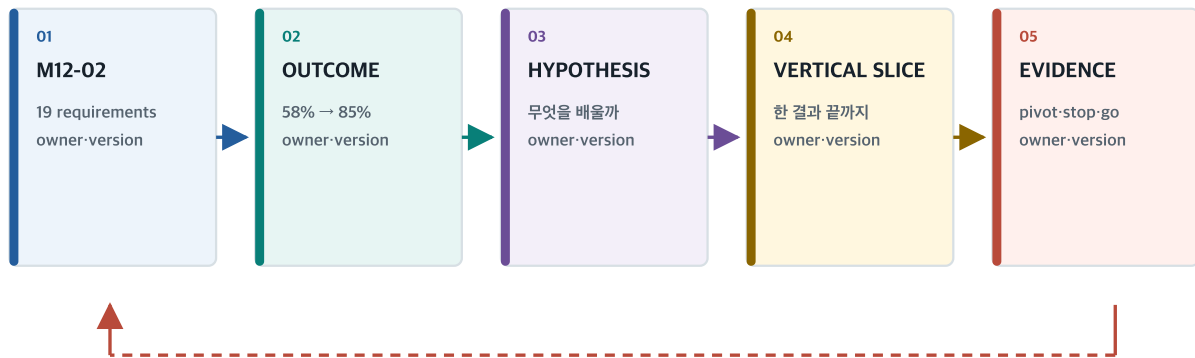
중요: 본 매뉴얼의 candidate는 실제 사용자·data·제품·예산·계약을 사용하지 않습니다. 합성 Gate를 통과해도 실제 실험과 release에는 별도의 윤리·법률·보안·접근성·운영·권한 검토가 필요합니다.

3. M12-02 요구를 outcome-learning 사슬로 바꾸기

M12-03 · 01

MVP는 기능 수가 아니라 학습 사슬로 정의합니다

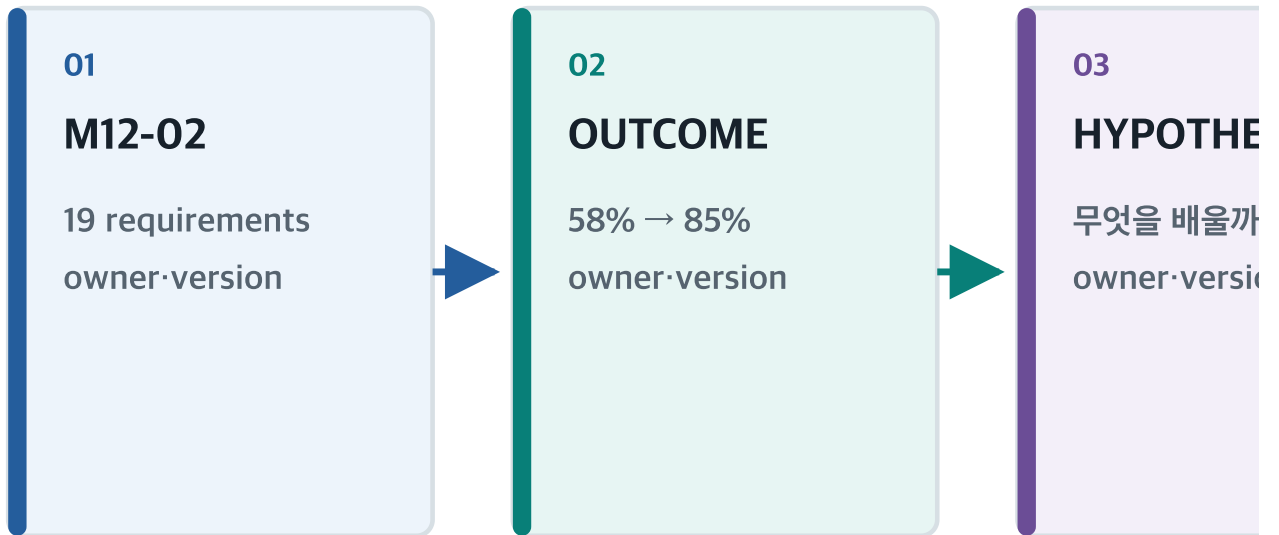
준비된 요구를 결과·가설·완전한 slice·증거·결정으로 연결하고 다시 갱신합니다.



최소 effort의 목적은 가장 중요한 불확실성을 줄여 다음 결정을 바꾸는 것입니다.

그림 1. 준비된 요구를 모두 만드는 것이 아니라 outcome과 가장 중요한 가설을 기준으로 한 완전한 slice와 evidence를 연결합니다.

준비된 요구를 결과·가설·완전한 slice·증거·결정으로 연결하고 다시 갱신함



니다.



CORE IDEA MVP의 ‘최소’는 기능 수가 아니라 결정에 필요한 학습 effort입니다.

M12-02는 무엇을 어떤 품질과 evidence로 완료할지 준비했습니다. M12-03은 그 19개 requirement와 4개 constraint를 전부 첫 버전에 넣지 않고, product goal과 primary hypothesis에 직접 필요한 범위를 선택합니다. 선택의 끝은 기능 출시가 아니라 어떤 evidence가 다음 결정을 바꾸는가입니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Input	M12-02 handoff v1	19 requirement·4 constraint·TBR 2
Outcome	완전 기록률 58%→85%	재입력 14→5분·오배정 ≤2%
Hypothesis	한 source + 담당·기한 확인	value·usability·feasibility·viability
Slice	create→confirm→status→correct→recover	UI·API·data·auth·error·evidence
Decision	persevere·pivot·stop	metric·threshold·owner·cadence

틀린 예와 고친 예

틀린 예: M12-02 requirement 19개를 중요도 점수로 정렬하고 위에서부터 개발합니다.

고친 예: Product goal과 primary hypothesis를 먼저 고르고, outcome을 관찰할 수 있는 vertical slice와 quality floor를 선택합니다.

5분 연습

1. M12-02 handoff version을 적습니다.
2. 이번 범위가 바꾸려는 outcome 하나를 씁니다.
3. 틀리면 방향이 가장 크게 바뀌는 hypothesis 하나를 씁니다.
4. 그 답으로 바뀔 결정을 한 줄로 씁니다.

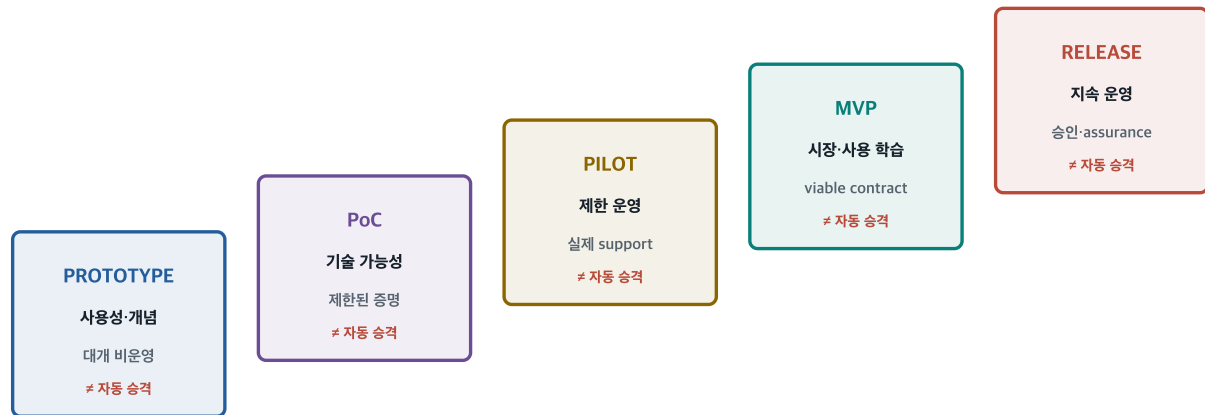
체크포인트: MVP의 '최소'는 기능 수가 아니라 결정에 필요한 학습 effort입니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

4. Prototype·PoC·pilot·MVP·release 경계 세우기

M12-03 · 02

Prototype·PoC·pilot·MVP·release는 같은 말이 아닙니다

목적·사용자·fidelity·운영·risk·승인 경계가 다르므로 artifact stage를 먼저 선언합니다.



앞 단계의 성공은 다음 단계 승인이나 공개 release를 자동으로 의미하지 않습니다.

그림 2. Artifact 단계는 fidelity 순서가 아니라 질문·대상·risk·운영·approval 계약이 다릅니다.

목적·사용자·fidelity·운영·risk·승인 경계가 다르므로 artifact stage를 먼저



선언합니다.

OT

운영

upport

승격

MVP

시장·사용 학습

viable contract

≠ 자동 승격

RELEASE

지속 운영

승인·assurance

≠ 자동 승격

CORE IDEA 앞 단계의 성공은 다음 단계 승인과 같지 않습니다.

Prototype은 주로 이해와 사용성을, PoC는 기술 가능성을, pilot은 제한 운영을, MVP는 실제 사용에서 제품 가설을, release는 지속 운영 책임을 다룹니다. 같은 화면이라도 어떤 stage인지에 따라 필요한 data·support·assurance·authority가 달라집니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Prototype	개념·flow·사용성	대개 비운영·합성 data
PoC	기술·integration 가능성	제한 환경·성능 범위
Pilot	제한 운영·support	대상·기간·data·risk 제한
MVP	제품 가설의 실제 학습	viable flow·metric·decision
Release	지속 제공	security·privacy·ops·approval

틀린 예와 고친 예

틀린 예: 클릭 가능한 prototype이 잘 작동하므로 MVP도 검증됐고 release해도 됩니다.

고친 예: Prototype evidence가 답한 질문과 답하지 못한 운영·risk 질문을 분리하고 다음 stage Gate를 별도로 둡니다.

5분 연습

1. 내 artifact의 stage 하나를 고릅니다.
2. 답할 질문과 답하지 못할 질문을 각각 씁니다.
3. 사용자·data·운영·approval 경계를 적습니다.
4. 다음 stage로 자동 승격되지 않는다고 명시합니다.

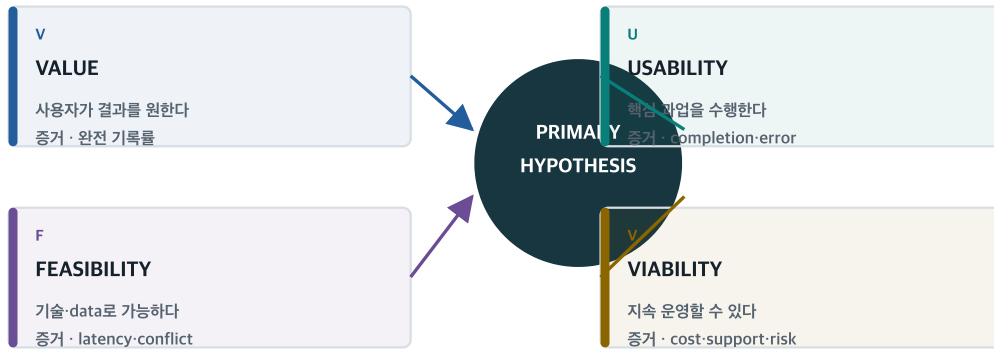
체크포인트: 앞 단계의 성공은 다음 단계 승인과 같지 않습니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

5. 가장 위험한 가설을 먼저 고르기

M12-03 · 03

가장 위험한 가정 하나가 MVP의 중심 질문이 됩니다

Value·usability·feasibility·viability를 분리하고 틀렸을 때 결정을 가장 크게 바꾸는 가정을 먼저 검증합니다.



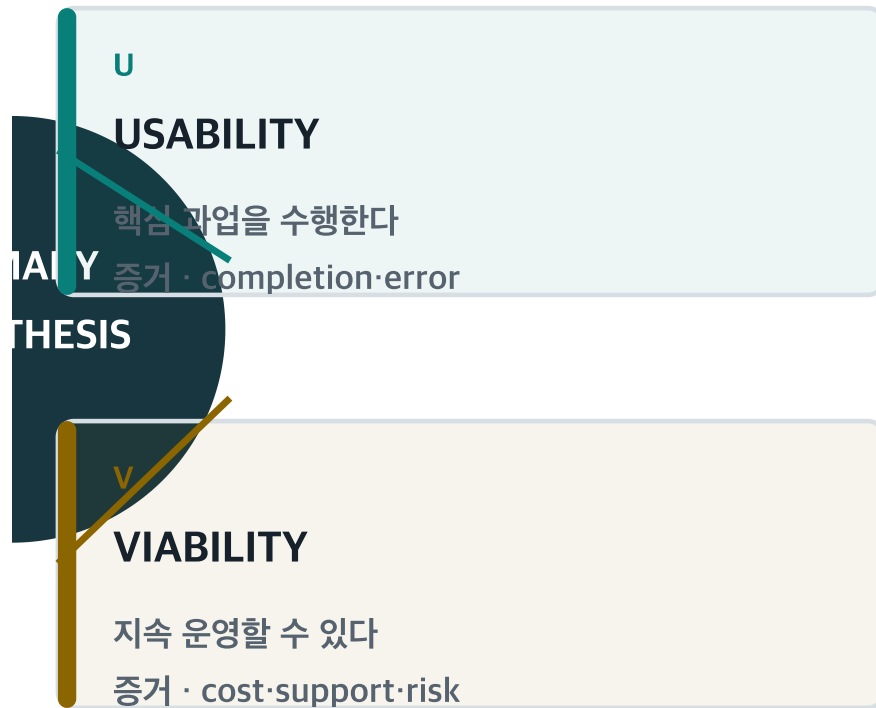
가설에는 대상·상황·행동·예상 변화·측정·반증·결정이 함께 있어야 합니다.

그림 3. Value·usability·feasibility·viability 가정을 분리한 뒤 불확실성과 영향이 가장 큰 질문을 먼저 검증합니다.

Value·usability·feasibility·viability를 분리하고 틀렸을 때 결정을 가장 크게



에 바꾸는 가정을 먼저 검증합니다.



CORE IDEA 쉬운 것을 먼저 만드는 대신 틀렸을 때 방향이 크게 바뀌는 것을 먼저 배웁니다.

Value는 사용자가 결과를 원하는지, usability는 과업을 수행하는지, feasibility는 기술과 data로 가능한지, viability는 비용·지원·정책·위험 안에서 지속 가능한지를 묻습니다. 한 가설에 네 질문을 섞으면 어떤 evidence가 무엇을 지지했는지 알 수 없습니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Value	한 source 기록을 선택하는가	실제 생성·확인 행동
Usability	담당·기한 확인을 완료하는가	completion·error·지원
Feasibility	충돌 없이 version을 보존하는가	latency·conflict·recovery
Viability	비용·support·privacy floor를 지키는가	cost·incident·manual work
Primary	완전률↑·재입력↓	답이 scope 결정을 바꿈

틀린 예와 고친 예

틀린 예: 사용자가 좋아할 것이다.

고친 예: 운영 조정 담당자가 한 회의 source에서 기록하고 담당·기한을 확인하면, 4주 동안 완전 기록률이 58%에서 85%로 높아지고 재입력이 14분에서 5분으로 줄어든다.

5분 연습

1. 네 assumption마다 한 문장을 씁니다.
2. 불확실성과 영향에 1~5를 줍니다.
3. 곱이 큰 하나를 primary로 고릅니다.
4. 반증 evidence와 바뀔 결정을 붙입니다.

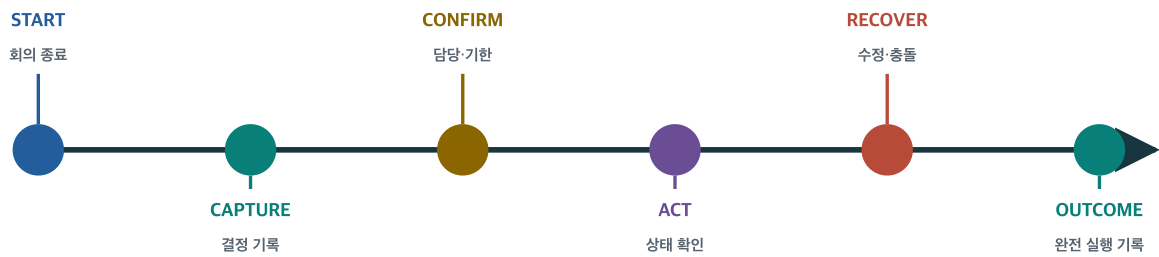
체크포인트: 쉬운 것을 먼저 만드는 대신 틀렸을 때 방향이 크게 바뀌는 것을 먼저 배웁니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

6. 사용자 문제의 시작과 끝을 끊지 않기

M12-03 · 04

MVP는 사용자 문제의 시작과 끝을 끊지 않습니다

Primary user가 실제 상황에서 한 결과를 달성하는 journey와 channel-support 경계를 함께 봅니다.

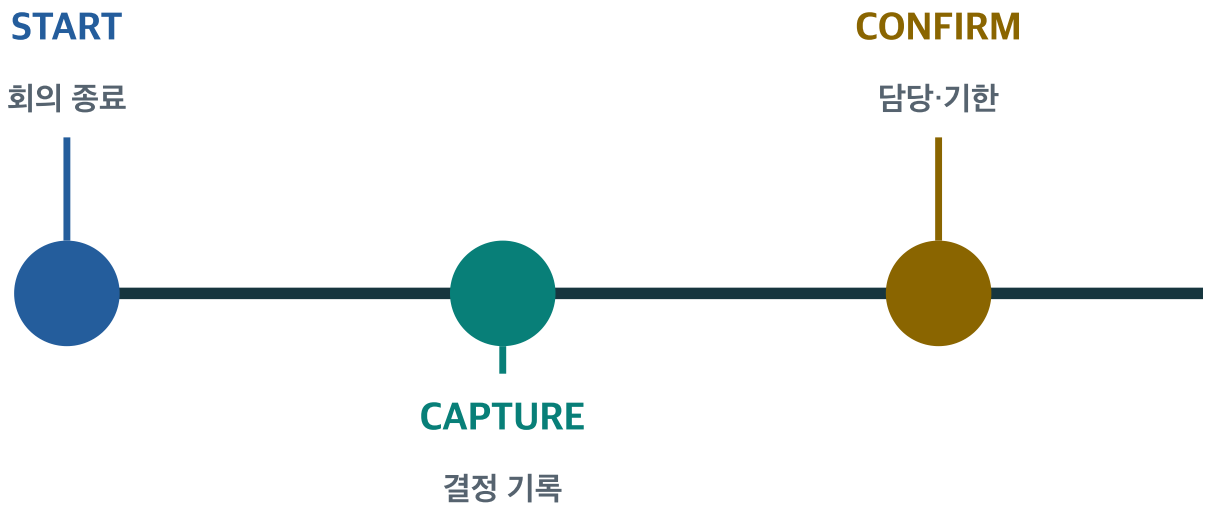


채널·support·접근성도 문제 경계 안에

기능은 적어도 사용자가 결과에 도달하고 오류에서 다시 일할 수 있어야 viable합니다.

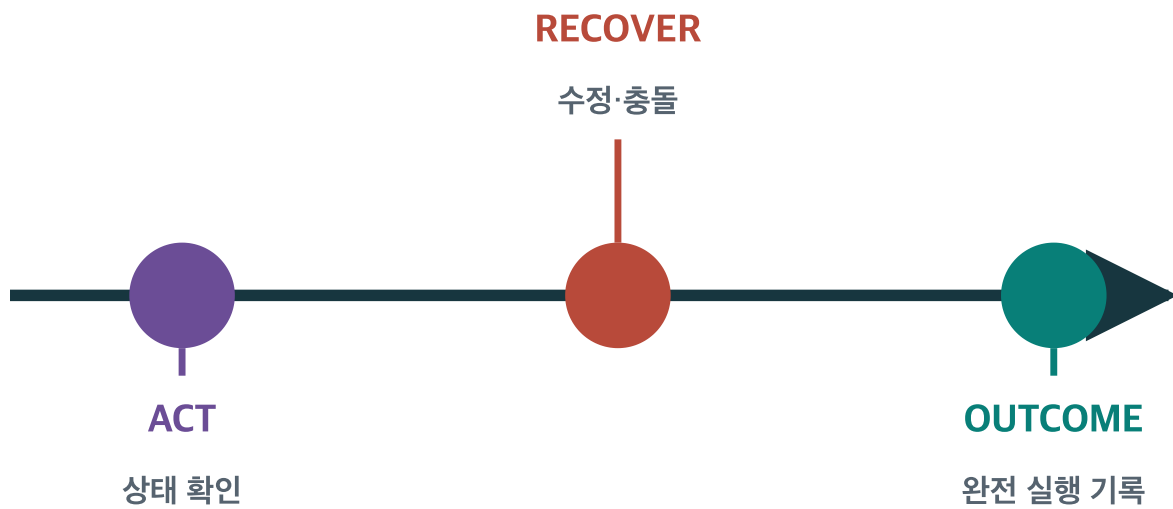
그림 4. 사용자는 기능 하나가 아니라 시작·확인·진행·오류·복구를 거쳐 결과를 얻습니다.

Primary user가 실제 상황에서 한 결과를 달성하는 journey와 channel·sup



채널·support·접근

oport 경계를 함께 봅니다.



성도 문제 경계 안에

CORE IDEA 작게 만든다는 이유로 사용자 문제의 가운데를 잘라서는 안 됩니다.

GOV.UK Service Standard는 service가 사용자의 whole problem을 해결해야 한다고 강조합니다. 합성 사례의 시작은 회의 종료이고 끝은 실행 가능한 기록입니다. 중간에 담당·기한 확인, 상태, 수정, 충돌 복구가 빠지면 작은 기능은 있어도 whole outcome은 없습니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Primary user	운영 조정 담당자	직접 기록 생성
Beneficiary	실행 담당 팀원	책임·기한 확인
Start	회의 종료	source 선택 가능
End	완전 실행 기록	owner·due·status 관찰
Boundary	한 source·제한 support	공개 rollout 제외

틀린 예와 고친 예

틀린 예: 회의 내용을 입력하는 화면만 MVP로 만듭니다.

고친 예: 회의 종료부터 create·confirm·status·correct·recover를 거쳐 완전 실행 기록에 도달하는 한 흐름을 MVP boundary로 둡니다.

5분 연습

1. Primary user와 beneficiary를 분리합니다.
2. Start event와 end outcome을 씁니다.
3. 중간 과업과 실패·복구를 한 줄로 잇습니다.
4. Channel·support·out-of-scope를 표시합니다.

체크포인트: 작게 만든다는 이유로 사용자 문제의 가운데를 잘라서 안 됩니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

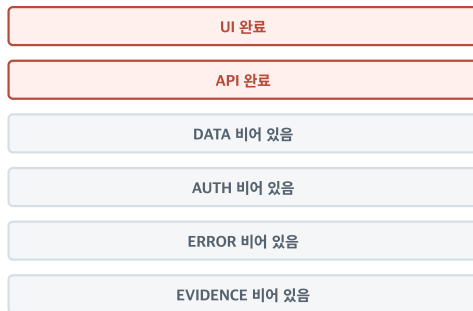
7. 가장 작은 end-to-end vertical slice 만들기

M12-03 · 05

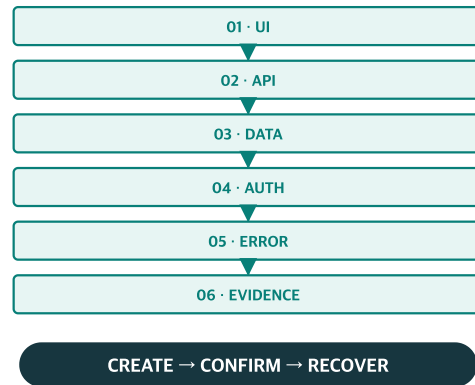
작은 MVP는 얇은 세로 조각입니다

UI만 또는 backend만 만드는 수평 절단이 아니라 UI·API·data·권한·오류·증거를 한 결과로 관통합니다.

HORIZONTAL · 막힘



VERTICAL · 결과



세로 slice는 작아도 end-to-end test·demo·측정이 가능하고 DoD를 만족해야 합니다.

그림 5. 작은 vertical slice는 기술 계층을 관통해 한 user outcome을 test·demo·measure할 수 있습니다.

UI만 또는 backend만 만드는 수평 절단이 아니라 UI·API·data·권한·오류·증거

HORIZONTAL · 막힘

UI 완료

API 완료

DATA 비어 있음

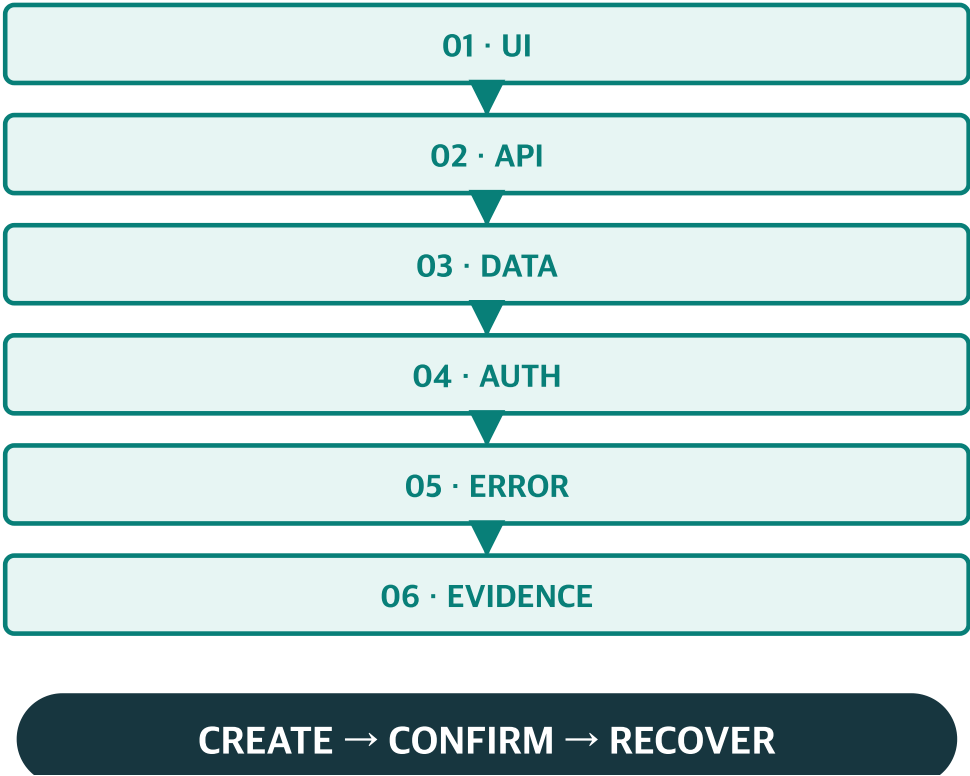
AUTH 비어 있음

ERROR 비어 있음

EVIDENCE 비어 있음

증거를 한 결과로 관통합니다.

VERTICAL · 결과



CORE IDEA 폭은 줄여도 깊이와 결과는 끊지 않습니다.

UI만 끝내거나 API만 넓게 만드는 horizontal slice는 integration과 실제 학습을 뒤로 미룹니다. Vertical slice는 사례 수와 변형을 줄이되 UI·API·data·authorization·error recovery·evidence를 모두 얹게 연결합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
UI	필수값·상태·오류	keyboard core flow
API	validate·authorize·respond	deny default
Data	draft·ready·version history	conflict 보존
Recovery	invalid·denied·concurrent	다시 일하기
Evidence	완전률·재입력·오배정	decision scorecard

틀린 예와 고친 예

틀린 예: 1주차 UI, 2주차 API, 3주차 DB, 4주차 통합으로 나눕니다.

고친 예: 첫 slice부터 create→confirm 한 흐름을 UI·API·data·권한·오류·metric으로 관통하고, 다음 slice에서 correction과 recovery를 확장합니다.

5분 연습

1. 한 user outcome을 고릅니다.
2. UI·API·data·authorization·error·evidence 여섯 줄을 씁니다.
3. 없어도 outcome이 유지되는 변형을 제거합니다.
4. end-to-end test와 DoD를 붙입니다.

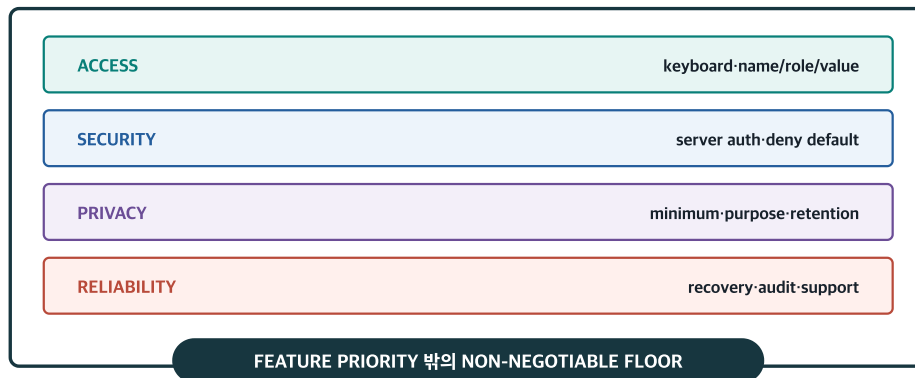
체크포인트: 폭은 줄여도 깊이와 결과는 끊지 않습니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

8. 품질·access·security·privacy·ops floor 보존하기

M12-03 · 06

품질 하한선은 우선순위 점수와 교환하지 않습니다

Access·security·privacy·performance·audit·recovery·support를 viable한 사용의 바닥으로 유지합니다.



‘MVPL니까 나중에’가 허용되는 항목과 처음부터 지켜야 할 floor를 명시적으로 분리합니다.

그림 6. 품질 하한선은 scope item 점수와 거래하는 feature가 아니라 viable 사용의 바닥입니다.

Access·security·privacy·performance·audit·recovery·support를 viable

ACCESS

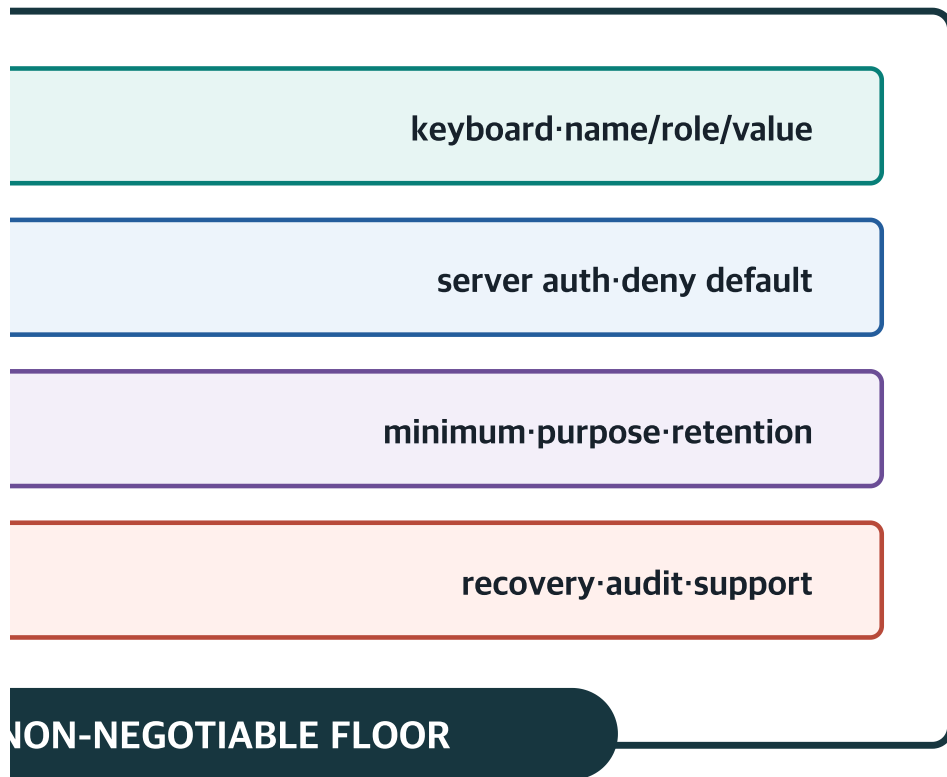
SECURITY

PRIVACY

RELIABILITY

FEATURE PRIORITY 밖의 N

한 사용의 바닥으로 유지합니다.



CORE IDEA MVP의 최소는 피해와 배제를 최소화한다는 뜻도 포함합니다.

접근성·권한·privacy·recovery를 나중으로 미루면 일부 사용자는 core flow를 수행할 수 없고, 초기 evidence는 안전하지 않은 조건에서 모입니다. M12-02의 품질 requirement를 별도 floor register로 유지해야 합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Access	keyboard·name/role/value	core flow 완료
Security	server authorization	허용 밖 노출 0
Privacy	목적 제한·최소 data	raw payload log 0
Performance	정한 load에서 p95	core flow threshold
Reliability	conflict·recovery·audit	silent overwrite 0

틀린 예와 고친 예

틀린 예: 보안·접근성·audit은 MVP가 성공한 뒤 hardening sprint에서 합니다.

고친 예: Core flow에 필요한 최소 floor를 now에 포함하고, 더 높은 assurance나 확대 적용은 별도 범위와 authority로 둡니다.

5분 연습

1. M12-02 quality requirement를 가져옵니다.
2. 없으면 user outcome이나 safety가 깨지는 floor를 표시합니다.
3. 측정 context·threshold·evidence를 붙입니다.
4. Feature score와 교환하지 않는다고 적습니다.

체크포인트: MVP의 최소는 피해와 배제를 최소화한다는 뜻도 포함합니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

9. Scope inventory·분해·명시적 제외 만들기

M12-03 · 07

범위는 포함 목록과 제외 계약을 함께 가집니다

모든 item을 requirement·journey·risk·owner·effort에 연결하고 now·next·later·not-now로 버전 관리합니다.

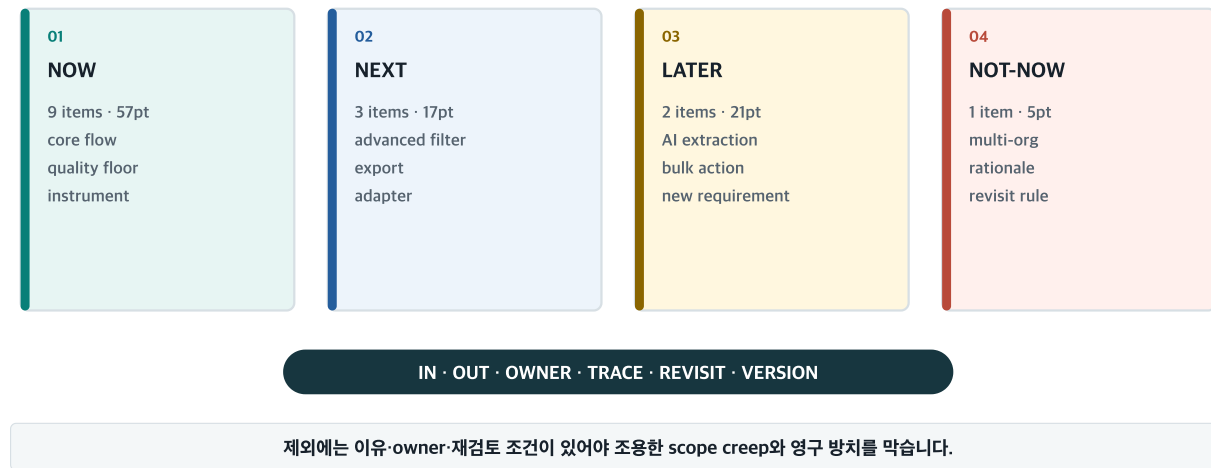


그림 7. 포함 목록만이 아니라 제외 이유와 revisit 조건까지 있어야 scope가 닫힙니다.

모든 item을 requirement·journey·risk·owner·effort에 연결하고 now·ne

01

NOW

9 items · 57pt

core flow

quality floor

instrument

02

NEXT

3 items · 17pt

advanced filter

export

adapter

IN · OUT · OWNER · TRA

ext·later·not-now로 버전 관리합니다.

03

LATER

2 items · 21pt

AI extraction

bulk action

new requirement

04

NOT-NOW

1 item · 5pt

multi-org

rationale

revisit rule

ACE · REVISIT · VERSION

CORE IDEA ‘나중에’는 bucket이 아니라 이유와 재검토 조건이 있는 계약입니다.

합성 candidate는 15개 scope item을 now 9·next 3·later 2·not-now 1로 나눕니다. AI 자동 추출·bulk action·multi-organization analytics·공개 rollout은 명시적으로 제외합니다. 조용히 빠진 항목과 의도적으로 제외한 항목을 구분해야 합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Now	9 item·57pt	core flow·floor·instrument
Next	3 item·17pt	filter·export·adapter
Later	2 item·21pt	AI·bulk, 새 requirement 필요
Not-now	1 item·5pt	multi-org analytics
Exclusion	4개	reason·owner·revisit

틀린 예와 고친 예

틀린 예: AI 기능은 일단 backlog 맨 아래에 두고 나중에 생각합니다.

고친 예: AI 자동 추출은 manual flow outcome 통과 뒤 별도 requirement와 risk review로 재검토한다고 적습니다.

5분 연습

1. Scope item을 requirement·journey 단위로 씁니다.
2. 각 item에 effort·risk·owner·dependency를 붙입니다.
3. 네 bucket에 배치합니다.
4. 제외 4개에 rationale·revisit를 씁니다.

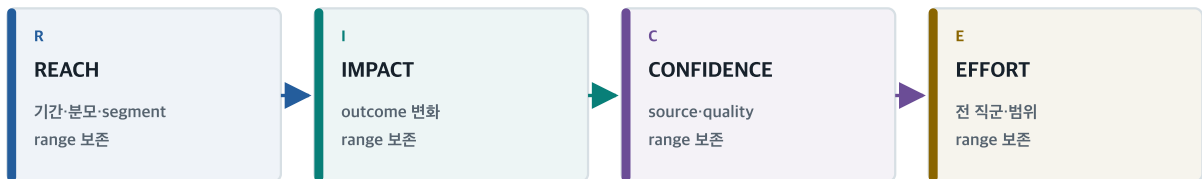
체크포인트: ‘나중에’는 bucket이 아니라 이유와 재검토 조건이 있는 계약입니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

10. Reach·impact·confidence·effort 원자료 보존하기

M12-03 · 08

우선순위 점수보다 원자료와 가정이 먼저입니다

RICE를 쓸 때 reach 기간·분모, impact outcome, confidence source, whole-team effort와 범위를 보존합니다.



$$\text{RICE} = (\text{reach} \times \text{impact} \times \text{confidence}) \div \text{effort}$$

score는 비교 신호 · 결정에는 risk·dependency·floor·strategy가 더 필요

점수가 높다는 사실만으로 Must·now·MVP를 결정하지 않습니다.

그림 8. RICE는 비교 신호이며 입력 source·기간·분모·범위·불확실성이 결정의 핵심 evidence입니다.

RICE를 쓸 때 reach 기간·분모, impact outcome, confidence source, wh



RICE = (reach × impact

score는 비교 신호 · 결정에는 risk·de

ole-team effort와 범위를 보존합니다.



× confidence) ÷ effort

dependency·floor·strategy가 더 필요

CORE IDEA 정밀한 점수보다 정직한 입력 범위가 더 중요합니다.

Intercom의 RICE는 reach·impact·confidence·effort를 사용해 아이디어를 비교하는 방식입니다. 하지만 reach 기간이나 denominator, impact가 연결된 outcome, confidence source, whole-team effort가 없으면 score theater가 됩니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Reach	4주·대상 meeting 수	기간·분모·segment
Impact	완전률·재입력 변화	outcome 연결
Confidence	M12-01·02 evidence	source quality
Effort	설계부터 support까지	range·uncertainty
Decision	score + risk + dependency	score 단독 금지

틀린 예와 고친 예

틀린 예: 점수 8.73이 가장 높으므로 무조건 now입니다.

고친 예: 입력 source와 range를 보여 주고, quality floor·critical dependency·risk·strategy를 함께 review해 now 여부를 결정합니다.

5분 연습

- 1. Reach의 기간·분모를 씁니다.
- 2. Impact가 바꾸는 outcome을 적습니다.
- 3. Confidence source를 직접·간접으로 구분합니다.
- 4. Whole-team effort range와 score sensitivity를 계산합니다.

체크포인트: 정밀한 점수보다 정직한 입력 범위가 더 중요합니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

11. 점수 밖의 risk·delay·opportunity cost 비교하기

M12-03 · 09

좋은 우선순위는 점수 밖의 손실과 위험을 보입니다

User value·risk reduction·cost of delay·opportunity cost·reversibility를 같은 의사결정 기록에서 비교합니다.



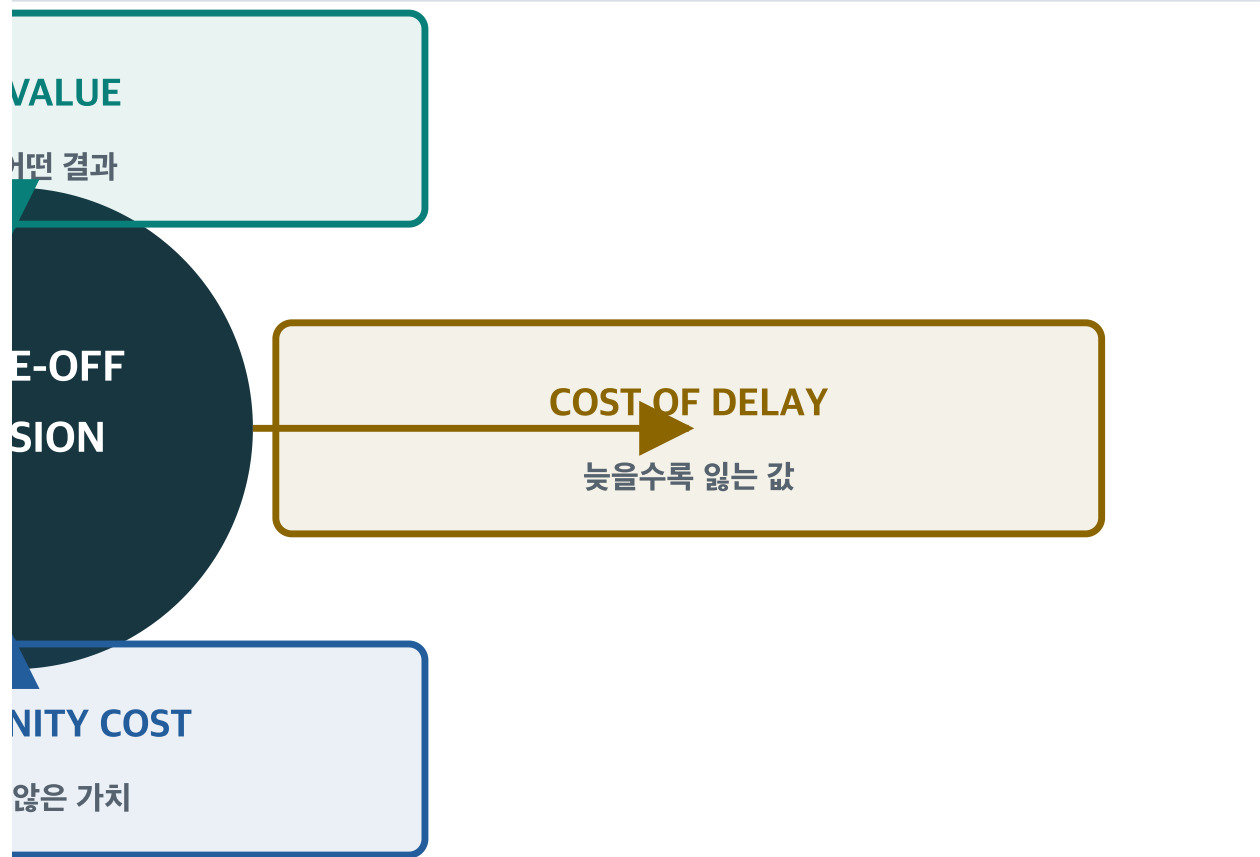
선택한 것뿐 아니라 포기한 것과 그 이유를 남겨야 다음 review에서 결정을 다시 계산할 수 있습니다.

그림 9. 우선순위는 선택한 가치뿐 아니라 자연 손실·피해 감소·포기한 대안·되돌릴 수 있는지를 함께 봅니다.

User value·risk reduction·cost of delay·opportunity cost·reversibility를



를 같은 의사결정 기록에서 비교합니다.



CORE IDEA 좋은 decision record는 선택하지 않은 것의 비용도 보여 줍니다.

Score가 비슷한 항목은 user harm·risk reduction·cost of delay·opportunity cost·reversibility에서 갑니다. 안전 floor는 score가 낮아도 반드시 필요하고, 되돌리기 어려운 범위는 더 강한 evidence와 authority가 필요합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
User value	완전 기록·재입력 감소	primary outcome
Risk reduction	오배정·노출·덮어쓰기 감소	guardrail
Cost of delay	누락이 계속 발생	시간별 손실
Opportunity	AI 대신 manual flow evidence	포기한 학습
Reversibility	scope bucket 변경 가능	release는 별도

틀린 예와 고친 예

틀린 예: 사용자가 많이 요청한 AI extraction을 가장 먼저 합니다.

고친 예: Manual create·confirm flow가 outcome을 만드는지 먼저 배우고, AI는 그 뒤 추가 value·risk·cost를 별도 가설로 검토합니다.

5분 연습

1. 후보 두 개를 고릅니다.
2. Value·risk·delay·opportunity·reversibility를 비교합니다.
3. 선택하지 않은 항목의 영향을 씁니다.
4. Decision owner와 review date를 붙입니다.

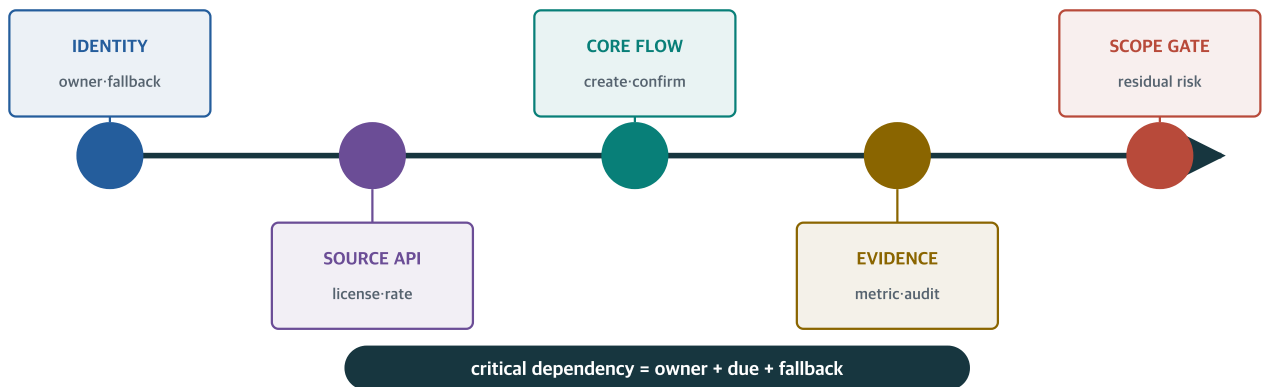
체크포인트: 좋은 decision record는 선택하지 않은 것의 비용도 보여 줍니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

12. Dependency graph와 critical sequence 만들기

M12-03 · 10

의존성은 목록이 아니라 순서와 대체 경로입니다

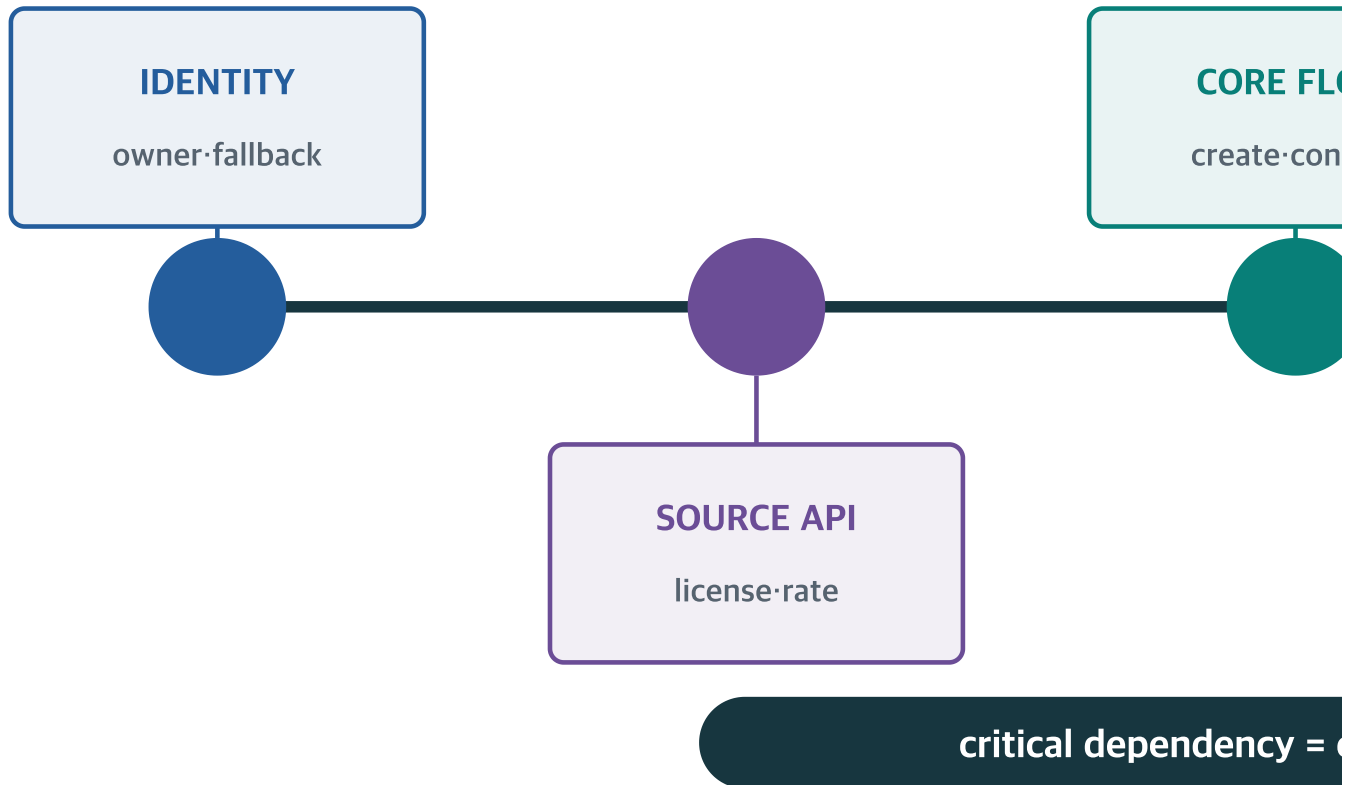
Identity·source·license·data·metric·approval dependency를 graph로 연결하고 critical sequence를 찾습니다.



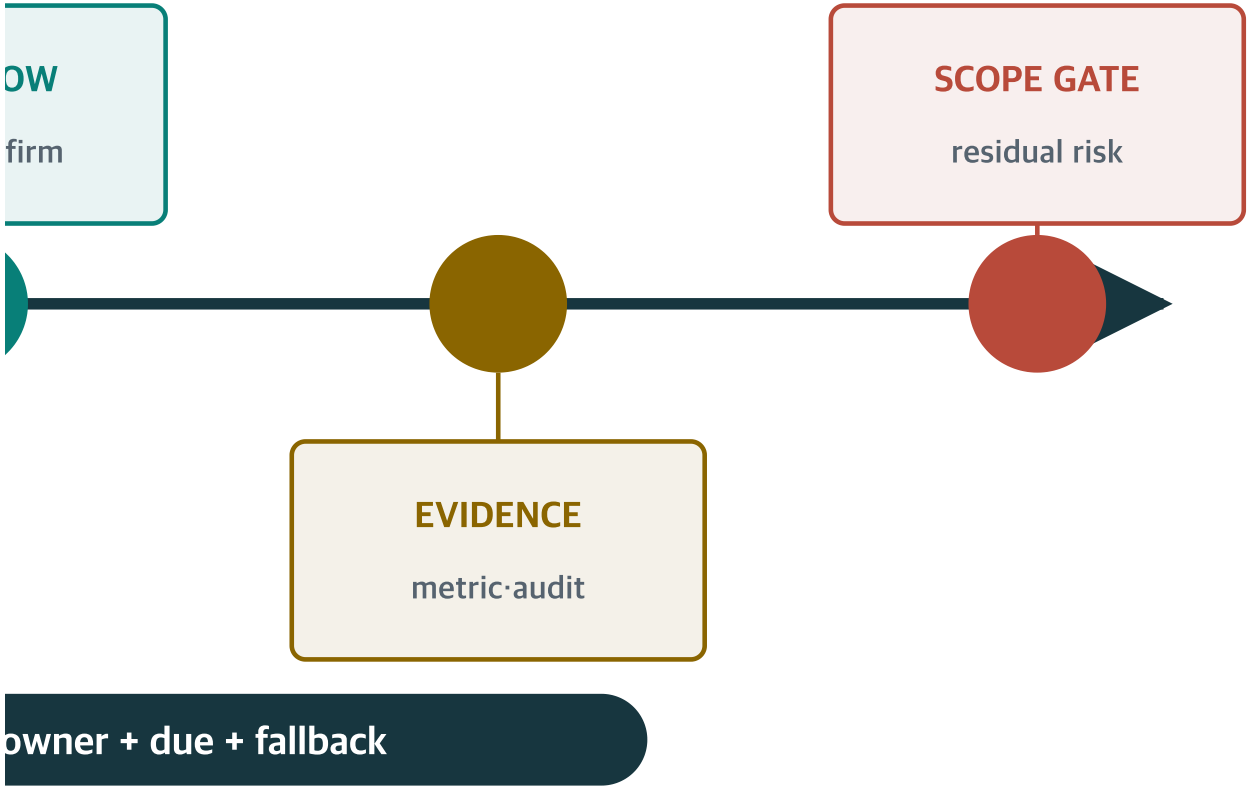
미해결 critical dependency가 1개라도 있으면 점수가 높아도 scope candidate를 멈춥니다.

그림 10. Dependency는 이름 목록이 아니라 predecessor·successor·owner·due·fallback이 있는 흐름입니다.

Identity·source·license·data·metric·approval dependency를 graph로



연결하고 critical sequence를 찾습니다.



CORE IDEA 이해결 critical dependency 하나는 높은 점수 여러 개보다 강한 중단 신호입니다.

합성 사례는 identity·meeting source·license·metric·approval 경계를 가집니다. '외부 API 필요'라고만 쓰면 언제 무엇이 막히는지 모릅니다. Graph로 연결해 critical sequence와 fallback을 정해야 capacity와 timebox가 현실적입니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Identity	조직·record 권한	owner·fallback
Source	meeting ID·link	API·license·rate
Core flow	create·confirm·recover	predecessor 완료
Evidence	event·audit·metric	Gate 입력
Critical open	0개	없지 않으면 blocked

틀린 예와 고친 예

틀린 예: Dependency: source API. 상태: 진행 중.

고친 예: Source API owner·due·required fields·rate·license·fallback manual link·영향 item·critical 여부를 연결합니다.

5분 연습

1. Internal·external dependency를 씁니다.
2. 선후 관계를 화살표로 잇습니다.
3. Owner·due·impact·fallback을 붙입니다.
4. Critical open 수를 계산합니다.

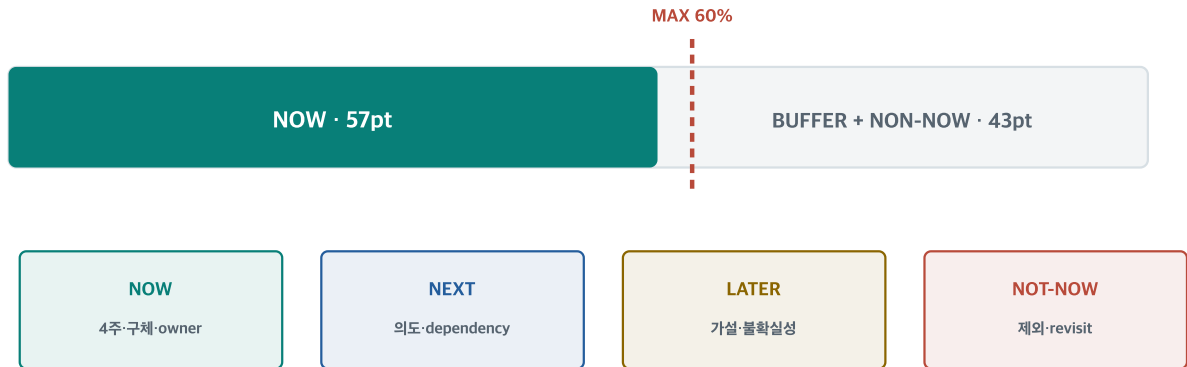
체크포인트: 미해결 critical dependency 하나는 높은 점수 여러 개보다 강한 중단 신호입니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

13. Now capacity·timebox·buffer 계약하기

M12-03 · 11

Now는 capacity 안에서 구체적으로, 먼 범위는 불확실하게 둡니다

합성 사례는 전체 100점 중 now 57점으로 60% 상한 안에 두고 나머지를 learning·dependency·buffer로 보존합니다.



Roadmap은 날짜 약속표가 아니라 outcome intent와 불확실성을 시간 방향으로 표현한 지도입니다.

그림 11. 합성 candidate는 now를 57점으로 제한하고 나머지를 buffer·non-now·learning·dependency에 남깁니다.

합성 사례는 전체 100점 중 now 57점으로 60% 상한 안에 두고 나머지를

NOW · 57pt

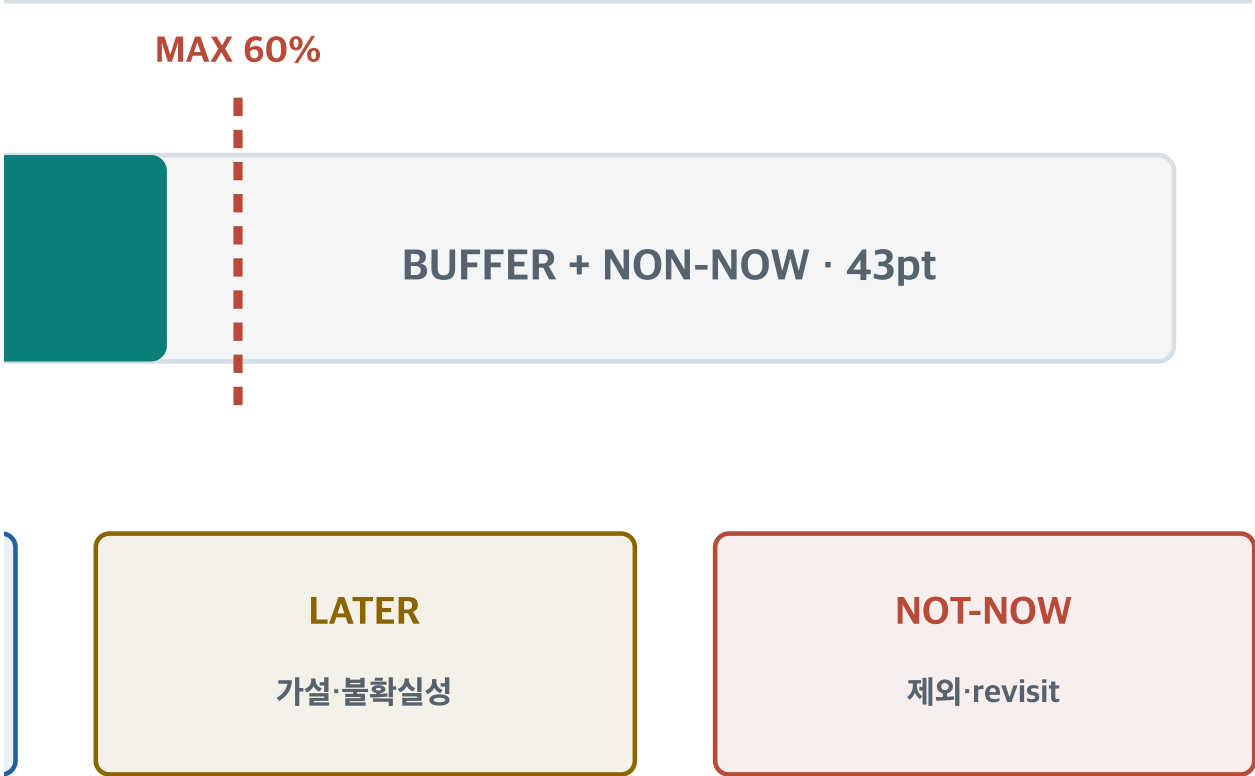
NOW

4주·구체·owner

NEXT

의도·dependency

learning·dependency·buffer로 보존합니다.



CORE IDEA 100% capacity 계획은 불확실한 MVP에서 이미 실패를 예약한 계획입니다.

Now는 4주 안에서 owner·DoD·evidence가 구체적이어야 합니다. Next와 later는 evidence가 쌓일수록 바뀔 수 있음을 드러냅니다. 합성 Gate는 now effort share를 60% 이하로 제한해 discovery·defect·support·dependency 변동을 흡수합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Capacity	100pt	4주 합성 timebox
Now	57pt	상한 60pt 이하
Next	17pt	now evidence 뒤
Later·not-now	26pt	새 가설·명시 제외
Buffer	계획에 내장	결함·지원·dependency

틀린 예와 고친 예

틀린 예: 전체 100점을 모두 now에 넣고 우선순위대로 최대한 합니다.

고친 예: Now 57점만 commitment 후보로 두고, 남은 범위와 buffer를 명시해 learning과 변동을 수용합니다.

5분 연습

1. 실제 capacity 산정 범위를 씁니다.
2. Now effort 합계를 계산합니다.
3. Quality floor가 빠지지 않았는지 봅니다.
4. 상한을 넘으면 slice 폭과 변형을 줄입니다.

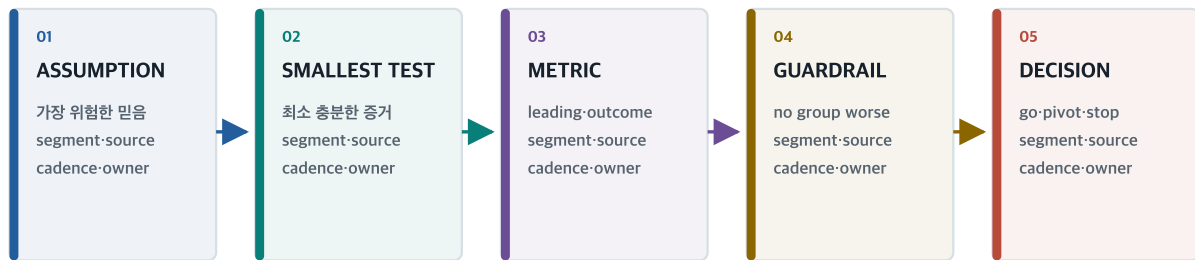
체크포인트: 100% capacity 계획은 불확실한 MVP에서 이미 실패를 예약한 계획입니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

14. Smallest sufficient test와 learning scorecard 연결하기

M12-03 · 12

MVP 범위에는 가장 작은 충분한 실험과 scorecard가 붙습니다

가정·test·leading·outcome·guardrail metric·segment·source·cadence·owner·반증을 한 줄로 연결합니다.



반증 증거를 먼저 쓰면 vanity metric을 피한다

실험은 기능을 증명하는 행사가 아니라 다음 결정에 필요한 불확실성을 줄이는 장치입니다.

그림 12. 가정별로 가장 작은 충분한 test와 leading·outcome·guardrail metric, 반증, decision을 한 줄로 연결합니다.



반증 증거를 먼저 쓰면 v

owner·반증을 한 줄로 연결합니다.



vanity metric을 피한다

CORE IDEA 실험은 성공을 보여 주는 demo가 아니라 방향을 바꿀 evidence를 얻는 장치입니다.

Value 가설에는 실제 선택 행동, usability에는 task completion과 error, feasibility에는 benchmark·conflict recovery, viability에는 cost·support·risk evidence가 필요합니다. 모든 가설에 같은 survey를 쓰면 evidence가 질문을 직접 답하지 못합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Value test	제한 flow 선택·사용	완전 기록물
Usability test	create·confirm task	completion·error·support
Feasibility test	load·conflict benchmark	p95·data loss
Viability test	cost·manual support 관찰	단위 비용·지원 시간
Guardrail	오배정·group·privacy	threshold 위면 stop

틀린 예와 고친 예

틀린 예: 만족도 5점 설문 하나로 네 가설을 모두 검증합니다.

고친 예: 각 가정에 직접 evidence를 주는 smallest test를 고르고, metric contract와 disconfirming evidence를 분리합니다.

5분 연습

- 1. Primary hypothesis를 다시 씁니다.
- 2. 가장 작은 충분한 test를 고릅니다.
- 3. Leading·outcome·guardrail metric을 정의합니다.
- 4. Segment·source·cadence·owner·limitation을 붙입니다.

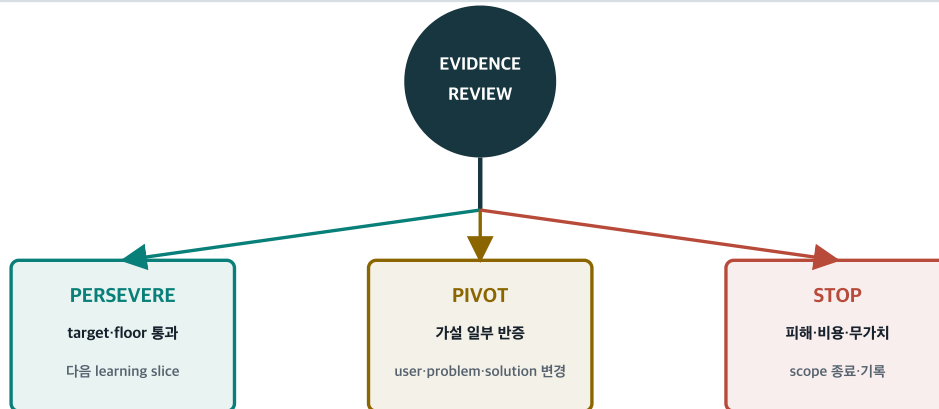
체크포인트: 실험은 성공을 보여 주는 demo가 아니라 방향을 바꿀 evidence를 얻는 장치입니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

15. Persevere·pivot·stop 규칙을 결과 전에 쓰기

M12-03 · 13

결정 규칙은 결과를 보기 전에 씁니다

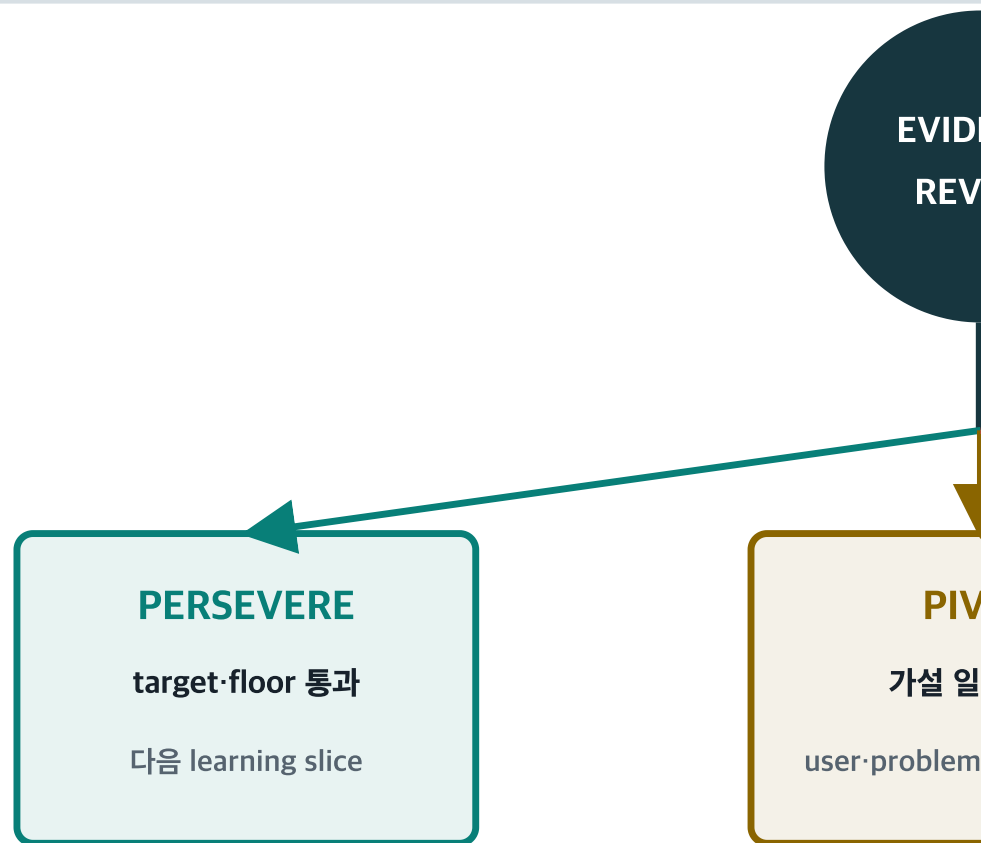
Persevere·pivot·stop의 threshold·review cadence·owner를 선등록해 좋은 소식만 고르는 판단을 막습니다.



어떤 결과도 자동 build·pilot·release 승인이 아니며 authority와 residual risk review가 별도로 필요합니다.

그림 13. 결과를 보기 전 threshold와 authority를 정해야 좋은 소식만 고르는 판단을 줄일 수 있습니다.

Persevere·pivot·stop의 threshold·review cadence·owner를 선등록해 줄



좋은 소식만 고르는 판단을 막습니다.



CORE IDEA Metric은 숫자를 모으기 위한 것이 아니라 행동을 바꾸기 위한 계약입니다.

Persevere는 target과 floor가 지지될 때 현재 방향을 유지하고 다음 slice를 정합니다. Pivot은 value·user·problem·solution·channel 가정 일부를 바꿉니다. Stop은 피해·비용·무가치·불가능 조건에서 scope나 test를 종료합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Persevere	완전률≥85%·floor 통과	다음 learning slice
Pivot	가치는 있으나 task 실패	flow·support 변경
Pivot	사용하지만 outcome 불변	problem·solution 재검토
Stop	오배정>2%·privacy incident	즉시 중단·review
Owner	product decision authority	주간 cadence·기록

틀린 예와 고친 예

틀린 예: 결과를 보고 팀이 회의에서 다음 방향을 정합니다.

고친 예: Threshold·segment·window·guardrail·owner를 사전에 정하고, 예외는 evidence와 authority로 기록합니다.

5분 연습

1. Persevere threshold를 씁니다.
2. 어떤 반증에서 무엇을 pivot할지 씁니다.
3. 즉시 stop 조건을 씁니다.
4. Review cadence·decision owner를 붙입니다.

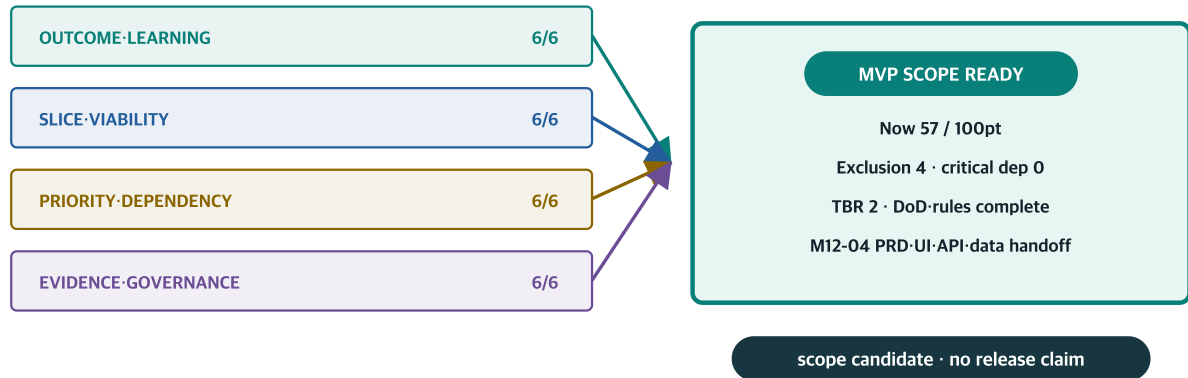
체크포인트: Metric은 숫자를 모으기 위한 것이 아니라 행동을 바꾸기 위한 계약입니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

16. MVP Scope Gate와 M12-04 handoff 닫기

M12-03 · 14

네 lane과 비기능 Gate가 모두 통과해야 범위 후보가 됩니다

24 scenario·12 control, outcome·slice·floor·priority·test 100%, now 57점, 제외 4, critical dependency 0을 함께 판정합니다.



mvp_scope_ready는 M12-04 문서 연결 후보이며 실제 MVP·pilot·release·예산·투자 승인이 아닙니다.

그림 14. 네 lane과 비기능 Gate를 모두 통과한 범위만 M12-04 문서 연결 후보가 됩니다.

24 scenario·12 control, outcome·slice·floor·priority·test 100%, now 5

OUTCOME·LEARNING	6/6
SLICE·VIABILITY	6/6
PRIORITY·DEPENDENCY	6/6
EVIDENCE·GOVERNANCE	6/6

7점, 제외 4, critical dependency 0을 함께 판정합니다.



MVP SCOPE READY

Now 57 / 100pt

Exclusion 4 · critical dep 0

TBR 2 · DoD·rules complete

M12-04 PRD·UI·API·data handoff

scope candidate · no release claim

CORE IDEA Gate는 ‘좋아 보인다’를 versioned evidence와 명시적 잔여 위험으로 바꿉니다.

합성 candidate는 24/24 scenario, critical 100%, 네 lane·12 control coverage 100%, outcome·hypothesis·vertical slice·quality floor·priority evidence·assumption test 100%, explicit exclusion 4, critical dependency 0, now 57점, TBR 2를 요구합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Outcome·learning	6/6	trace·goal·hypothesis·metric
Slice·viability	6/6	whole problem·floor·DoD
Priority·dependency	6/6	evidence·trade-off·capacity
Evidence·governance	6/6	test·rules·version·risk
Handoff	M12-04	PRD·screen·API·data trace

틀린 예와 고친 예

틀린 예: `mvp_scope_ready` 이므로 build와 pilot을 시작합니다.

고친 예: 합성 scope candidate를 M12-04 문서에 연결하고, 실제 build·pilot·release·예산·assurance는 권한 있는 별도 Gate로 둡니다.

5분 연습

1. 네 lane과 비기능 Gate를 판정합니다.
2. Fail·TBR·residual risk를 숨기지 않습니다.
3. Scope version과 authority를 기록합니다.
4. M12-04 문서 trace ID를 준비합니다.

체크포인트: Gate는 '좋아 보인다'를 versioned evidence와 명시적 잔여 위험으로 바꿉니다. 지금 쓴 항목의 source·owner·version·limitation을 다시 확인합니다.

17. 합성 candidate의 15개 scope item

아래 표는 점수 순 목록이 아닙니다. M12-02 requirement trace, whole journey, quality floor, dependency, learning evidence를 함께 만족하도록 bucket을 정한 결과입니다.

ID	Bucket	Effort	Scope item	Requirement trace	Dependency
SCP-01	now	8	한 source 실행 기록 생성·필수값 검증	FR-01, FR-02	none
SCP-02	now	8	담당·기한 확인과 수정 제안	FR-03	none
SCP-03	now	5	확인 상태와 현재 회의 기본 보기	FR-04, FR-06	none
SCP-04	now	8	권한 있는 수정과 version history	FR-05, QR-04, QR-07	none
SCP-05	now	5	동시 변경 충돌 감지·복구	FR-07	none
SCP-06	now	5	완전률·재입력·오배정 측정	QIU-01, QIU-02, QIU-03	none
SCP-07	now	8	접근성·authorization·privacy floor	QR-02, QR-03, QR-04	none
SCP-08	now	5	핵심 flow 성능 threshold	QR-01	none
SCP-09	now	5	제한 pilot reliability·recovery evidence	QR-05, QR-06	none
SCP-10	next	5	meeting·owner·due·status 고급 filter	FR-06	none
SCP-11	next	4	접근 가능한 export	FR-08	none
SCP-12	next	8	추가 meeting source adapter	C-04	source-api
SCP-13	later	13	AI 결정·담당·기한 추출	needs-requirement	none
SCP-14	later	8	회의 간 bulk 후속 조치	needs-requirement	none
SCP-15	not-now	5	다중 조직 analytics·ranking	excluded	none

17.1 네 개의 명시적 제외

ID	제외	Rationale	Revisit
EX-01	AI 자동 추출	핵심 확인 가정을 먼저 검증	manual flow target 통과 뒤 별도 requirement

ID	제외	Rationale	Revisit
EX-02	Bulk cross-meeting action	단일 meeting vertical slice 우선	반복 사용 evidence 뒤
EX-03	Multi-organization analytics	primary user outcome에 직접 필요하지 않음	MVP outcome 통과 뒤
EX-04	Public·production rollout	합성 scope 후보이며 실제 assurance·approval 없음	권한 있는 별도 release Gate

AI 자동 추출은 ‘좋지 않은 기능’이어서 빠진 것이 아닙니다. Manual create·confirm flow의 value와 usability를 먼저 배우는 것이 현재 decision utility가 더 크기 때문에 별도 requirement 뒤로 보냅니다.

18. 열두 개 MVP 범위 통제

통제는 순서대로 읽되 실제 review에서는 control ID로 scenario·evidence를 양방향 추적합니다.

Control	통제 제목	최소 evidence
CTRL-01	Product goal·outcome·requirement trace	Outcome-to-scope trace
CTRL-02	Riskiest hypothesis·learning decision	Hypothesis and decision card
CTRL-03	User·situation·whole·problem boundary	Whole-problem scope map
CTRL-04	Prototype·PoC·pilot·MVP·release boundary	Artifact-stage decision matrix
CTRL-05	Smallest end-to-end vertical slice	Vertical slice map
CTRL-06	Quality·access·security·privacy·ops floors	Viability floor register
CTRL-07	Scope inventory·decomposition·exclusion	Versioned scope table
CTRL-08	Reach·impact·confidence·effort evidence	Priority evidence sheet
CTRL-09	Risk·delay·opportunity cost·dependency	Risk and dependency map
CTRL-10	Now·next·later·not-now·capacity	Outcome roadmap and capacity contract
CTRL-11	Assumption test·metric·decision rules	Learning scorecard

Control	통제 제목	최소 evidence
CTRL-12	Version·DoD·residual risk·M12-04 handoff	MVP Scope Gate

19. 24개 합성 scenario portfolio

각 lane에는 정확히 6개 scenario가 있습니다. PO·critical scenario 하나라도 실패하면 높은 평균 점수와 무관하게 candidate는 blocked입니다.

ID	Lane	Scenario	Risk	Source → Sink	Control
SC-01	outcome-learning	M12-02 requirement·constraint·TBR trace	feature-pile	m12-02-handoff → scope-trace	CTRL-01
SC-02	outcome-learning	Product goal·desired outcome·guardrail 범위	feature-pile	problem-outcome-contract → product-goal	CTRL-01, CTRL-03
SC-03	outcome-learning	가장 위험한 hypothesis와 바뀌는 결정	feature-pile	assumption-register → learning-question	CTRL-02
SC-04	outcome-learning	Prototype·PoC·pilot·MVP·release 구분	false-mvp-approval	artifact-options → stage-boundary	CTRL-04
SC-05	outcome-learning	Primary user·situation·whole problem 시작과 끝	minimum-without-viable	current-journey → whole-problem-map	CTRL-03
SC-06	outcome-learning	Learning metric·baseline·target·segment·cadence	score-theater	outcome-contract → learning-measure	CTRL-01, CTRL-11
SC-07	slice-viability	UI·API·data·authorization·error·evidence vertical slice	minimum-without-viable	scope-candidates → vertical-slice	CTRL-05
SC-08	slice-viability	Create·confirm·status·correct·recover core job	minimum-without-viable	user-journey → core-job-slice	CTRL-03, CTRL-05
SC-09	slice-viability	Access·security·privacy·performance·audit·recovery floor	quality-floor-cut	m12-02-quality-requirements → viability-floor	CTRL-06

ID	Lane	Scenario	Risk	Source → Sink	Control
SC-10	slice-viability	Core flow usability·access support·channel handoff	quality-floor-cut	access-guardrail → inclusive-service-boundary	CTRL-03, CTRL-06
SC-11	slice-viability	Not-now 제외·rationale·revisit condition	feature-pile	scope-inventory → exclusion-register	CTRL-07
SC-12	slice-viability	Usable increment·Definition of Done·no approval claim	false-mvp-approval	vertical-slice → usable-increment	CTRL-05, CTRL-06, CTRL-12
SC-13	priority-dependency	Scope item·requirement·journey·risk inventory	feature-pile	m12-02-register → scope-inventory	CTRL-07
SC-14	priority-dependency	Must-have test와 now capacity 60% 이내	score-theater	scope-inventory → capacity-contract	CTRL-07, CTRL-10
SC-15	priority-dependency	Reach·impact·confidence·effort raw evidence	score-theater	priority-candidates → priority-evidence-sheet	CTRL-08
SC-16	priority-dependency	Risk reduction·cost of delay·opportunity cost	score-theater	risk-register → tradeoff-record	CTRL-09
SC-17	priority-dependency	Dependency owner·due·critical sequence	dependency-blindness	dependency-register → sequence-map	CTRL-09
SC-18	priority-dependency	Effort range·capacity·4주 timebox·buffer	dependency-blindness	effort-estimates → timebox-plan	CTRL-08, CTRL-10
SC-19	evidence-governance	가정별 smallest sufficient test·반증	false-mvp-approval	assumption-map → test-cards	CTRL-02, CTRL-11
SC-20	evidence-governance	Leading·outcome·guardrail scorecard	score-theater	metric-contract → learning-scorecard	CTRL-11
SC-21	evidence-governance	Persevere·pivot·stop rule·review cadence	false-mvp-approval	learning-scorecard → decision-rules	CTRL-02, CTRL-11
SC-22	evidence-governance	Now·next·later·not-now outcome roadmap	feature-pile	scope-baseline → outcome-roadmap	CTRL-07, CTRL-10
SC-23	evidence-governance	Scope version·change impact·TBR·residual risk	dependency-blindness	scope-evidence-pack → decision-log	CTRL-12
SC-24	evidence-governance	MVP Scope Gate·M12-04 handoff	false-mvp-approval	scope-evidence-pack → mvp-scope-gate	CTRL-01, CTRL-11, CTRL-12

20. 세 범위 version을 비교해 배우기

관찰	기능 수만 줄인 기준선	점수만 쓴 중간안	Evidence-led candidate
Scenario	6/24	15/24	24/24
Now effort	88/100	74/100	57/100
Outcome trace	20%	86%	100%
Vertical slice	17%	75%	100%
Quality floor	0%	62%	100%
Explicit exclusion	0	2	4
Critical dependency	3	1	0
Decision	blocked_feature_cut	blocked_score_theater	mvp_scope_ready

20.1 기준선이 실패하는 이유

기능명은 작아 보이지만 outcome trace·hypothesis·vertical depth·quality floor·exclusion·dependency·test·decision rule이 없습니다. Now effort도 88점이라 변동을 흡수할 여지가 없습니다.

20.2 중간안이 실패하는 이유

RICE 표와 bucket은 있지만 confidence source·uncertainty·quality floor·critical dependency·stop rule·M12-04 handoff가 불완전합니다. 숫자가 있다는 사실과 decision readiness는 다릅니다.

20.3 Candidate가 통과하는 이유

기능을 더 많이 넣어서가 아니라 outcome·hypothesis·whole problem·vertical slice·floor·raw priority evidence·dependency·capacity·test·decision·version이 하나의 evidence chain으로 닫혔기 때문입니다.

21. MVP 범위·우선순위 스튜디오 읽기

YONCORE · M12-03 SYNTHETIC LAB

MVP 범위·우선순위 스튜디오

합성 범위 후보 검증 · 실제 MVP 공개 pilot release 예전 후지 계획 인증 build 승인

01 M12-03 trace

02 goal hypothesis

03 vertical slice floor

04 priority dependency

05 test decision rule

06 MVP Scope Gate

Scope Item

157case

Now / next / later / not now

9 / 3 / 2 / 1

Now effort / capacity

57 / 100pt

Outcome hypothesis trace

100% / 100%

Quality flow / critical dependency

100% / 0

합성 case version

team-follow-up-mvp-scope-2026-07-v1

07 12개 범위 통제

08 24개 합성 MVP 시나리오

09 검증 과정

07 12개 범위 통제

08 24개 합성 MVP 시나리오

09 검증 과정

07 12개 범위 통제

08 24개 합성 MVP 시나리오

09 검증 과정

07 12개 범위 통제

08 24개 합성 MVP 시나리오

09 검증 과정

07 12개 범위 통제

08 24개 합성 MVP 시나리오

09 검증 과정

07 12개 범위 통제

08 24개 합성 MVP 시나리오

09 검증 과정

실습 화면 1. 왼쪽은 12 control과 6 failure type, 가운데는 24 scenario, 오른쪽은 scope version·summary·Gate·trace입니다.

YONCORE · GIBALJA IT-AI SERVICE DEVELOPMENT ROADMAP

M12-03 · 85 / 99

21.1 한 행을 evidence contract로 읽기

02 24개 합성 MVP 시나리오

전체

결과 학습

범위 viability

우선순위 의존성

중가 가버넌스

ID	영역	시험	입력 근거	범위 산출물	결과
SC-01	결과 학습	가능 대미	m12-02-handoff	scope-trace	PASS
SC-02	결과 학습	가능 대미	problem-outcome-contract	product-goal	PASS
SC-03	결과 학습	가능 대미	assumption-register	learning-question	PASS
SC-04	결과 학습	거짓 MVP 승인	artifact-options	stage-boundary	PASS
SC-05	결과 학습	올 수 있는 최소	current-journey	whole-problem-map	PASS
SC-06	결과 학습	완수 연극	outcome-contract	learning-measure	PASS
SC-07	범위 viability	올 수 없는 최소	scope-candidates	vertical-slice	PASS
SC-08	범위 viability	올 수 없는 최소	user-journey	core-job-slice	PASS
SC-09	범위 viability	품질 floor 절단	m12-02-quality-requirements	viability-floor	PASS
SC-10	범위 viability	품질 floor 절단	access-guardrail	inclusive-service-boundary	PASS
SC-11	범위 viability	가능 대미	scope-inventory	exclusion-register	PASS
SC-12	범위 viability	거짓 MVP 승인	vertical-slice	usable-increment	PASS
SC-13	우선순위 의존성	가능 대미	m12-02-register	scope-inventory	PASS
SC-14	우선순위 의존성	완수 연극	scope-inventory	capacity-contract	PASS
SC-15	우선순위 의존성	완수 연극	priority-candidates	priority-evidence-sheet	PASS
SC-16	우선순위 의존성	완수 연극	risk-register	tradeoff-record	PASS
SC-17	우선순위 의존성	의존성 행명	dependency-register	sequence-map	PASS
SC-18	우선순위 의존성	의존성 행명	effort-estimates	timebox-plan	PASS
SC-19	중가 가버넌스	거짓 MVP 승인	assumption-map	test-cards	PASS
SC-20	중가 가버넌스	완수 연극	metric-contract	learning-scorecard	PASS
SC-21	중가 가버넌스	거짓 MVP 승인	learning-scorecard	decision-rules	PASS
SC-22	중가 가버넌스	가능 대미	scope-baseline	outcome-roadmap	PASS
SC-23	중가 가버넌스	의존성 행명	scope-evidence-pack	decision-log	PASS
SC-24	중가 가버넌스	거짓 MVP 승인	scope-evidence-pack	mvp-scope-gate	PASS

MVP scope evidence contract

SC-14: Must-have test의 now capacity 60% 이내

BOUNDARY · 범위 경계

```
{
  "zone": "delivery-slice",
  "control": {
    "CTRL-Q": "",
    "CTRL-18"
  }
}
```

EXPECTED · readiness oracle

```
{
  "code": "MUST_CAPACITY_BALANCED",
  "maximum_now_effort_points": 60,
  "must_have_test_present": true,
  "now_effort_points": 43,
  "now_effort_points": 57,
  "quality_floor_not_traded": true
}
```

FLOW · 근거에서 한행

```
{
  "source": "scope-inventory",
  "sink": "capacity-contract",
  "lane": "priority-dependency"
}
```

ACTUAL · 현재 version

```
{
  "code": "MUST_CAPACITY_BALANCED",
  "maximum_now_effort_points": 60,
  "must_have_test_present": true,
  "now_effort_points": 43,
  "now_effort_points": 57,
  "quality_floor_not_traded": true
}
```

내 lane 검증률

24/24 scenario evidence

결과 학습

범위 viability

우선순위 의존성

중가 가버넌스

6/6

6/6

6/6

6/6

실습 화면 2. Scenario를 선택하면 boundary·source→sink·expected oracle·actual을 비교할 수 있습니다.

1. `small-feature-cut-v1` 을 실행해 18개 FAIL을 유형별로 봅니다.
2. `score-only-priority-v2` 로 바꿔 점수표가 해결하지 못한 9개 FAIL을 봅니다.
3. `evidence-led-mvp-v3` 에서 24/24와 now 57점·exclusion 4·critical dependency 0을 확인합니다.
4. 기준선 비교에서 fixed 18·regressed 0을 확인합니다.
5. 6개 regression contract를 실행합니다.

21.2 모바일에서 세 구역 복습하기

① 통제-위험 통제

mvp-scope-controls-1.0.0 · mvp-scope-scenarios-1.0.0

CTRL-01

Product goal-outcome-requirement trace

M12-02의 19개 requirement-4개 constraint를 product goal-desired outcome-guardrail source version에 연결한다.

CTRL-02

Riskiest hypothesis-learning decision

가장 위험한 value-usability-feasibility-viability 가정과 팀이 바꾸는 persevere-pivot-stop 결정을 명시한다.

CTRL-03

User-situation-whole-problem boundary

Primary user-beneficiary-situation-start/end channel-support를 사용자 관점의 일관된 문제 범위로 정한다.

CTRL-04

Prototype-PoC-pilot-MVP-release boundary

학습 artifact의 복제 사용자 fidelity-risk-운영 approval을 구분하고 서로 같은 형태로 쓰지 않는다.

CTRL-05

Smallest end-to-end vertical slice

UI/API/data authorization-error-evidence를 가리지 않고 사용자가 한 결과를 끝까지 달성하는 최소 slice를 정한다.

CTRL-06

Quality-access-security-privacy-ops floors

M12-02의 측정 품질과 access-security-privacy-audit-recovery floor를 가장 우선순위와 별도로 non-negotiable하게 유지한다.

CTRL-07

Scope inventory-decomposition-exclusion

Scope item을 requirement-journey step-owner에 연결하고 now-next-later-not-now 및 명시적 제외 재검토 조건을 둔다.

CTRL-08

Reach-impact-confidence-effort evidence

RICE 중 scoring을 할 때 기간-문도-outcome-impact-confidence source-전 역문 effort와 불확실성 범위를 보른다.

CTRL-09

Risk-delay-opportunity cost-dependency

② 시나리오-evidence

SC-05	결과-학습	쓸 수 없는 최소	current-jour
SC-06	결과-학습	점수 연극	outcome-cc
SC-07	범위-viability	쓸 수 없는 최소	scope-candi
SC-08	범위-viability	쓸 수 없는 최소	user-journe
SC-09	범위-viability	품질 floor 절단	m12-02-qu
SC-10	범위-viability	품질 floor 절단	access-guar
SC-11	범위-viability	기능 더미	scope-inver
SC-12	범위-viability	거짓 MVP 승인	vertical-slic
SC-13	우선순위-의존성	기능 더미	m12-02-reg
SC-14	우선순위-의존성	점수 연극	scope-inver
SC-15	우선순위-의존성	점수 연극	priority-can
SC-16	우선순위-의존성	점수 연극	risk-register
SC-17	우선순위-의존성	의존성 행렬	dependency
SC-18	우선순위-의존성	의존성 행렬	effort-estim
SC-19	중거-거버넌스	거짓 MVP 승인	assumption
SC-20	중거-거버넌스	점수 연극	metric-cont
SC-21	중거-거버넌스	거짓 MVP 승인	learning-scc
SC-22	중거-거버넌스	기능 더미	scope-basel
SC-23	중거-거버넌스	의존성 행렬	scope-evide
SC-24	중거-거버넌스	거짓 MVP 승인	scope-evide

MVP scope evidence contract

행을 선택하세요

③ Gate-추적

기준선과 후보 비교

6개 회귀 계약

mvp_scope_ready

시나리오

24/24

Critical

100%

통제

12/12

영역

4/4

통제 추적

CTRL → scenario → evidence

CTRL-01	4 case · 0 fail	LINK
CTRL-02	3 case · 0 fail	LINK
CTRL-03	4 case · 0 fail	LINK
CTRL-04	1 case · 0 fail	LINK
CTRL-05	3 case · 0 fail	LINK
CTRL-06	3 case · 0 fail	LINK
CTRL-07	4 case · 0 fail	LINK
CTRL-08	2 case · 0 fail	LINK
CTRL-09	2 case · 0 fail	LINK
CTRL-10	3 case · 0 fail	LINK
CTRL-11	5 case · 0 fail	LINK
CTRL-12	3 case · 0 fail	LINK

실행 기록

1. evidence-led-mvp-v3: 24/24 PASS · now 57pt · exclusion 4 · mvp_scope_ready

2. CTRL 12개와 scenario 24개를 불러왔습니다.

3. 항상 MVP 범위 계약을 불러옵니다.

세 이미지를 왼쪽부터 보며 통제 → scenario → expected/actual → Gate → trace 를 말로 설명합니다.



실습 화면 3. 전체 모바일 흐름에서도 문서 순서와 동일하게 통제·scenario·Gate가 이어집니다.

22. 90분 실습 workbook

시간	행동	남길 evidence	통과 질문
0~10분	M12-02 handoff·goal·outcome 고정	source version·baseline·target	어떤 결과를 왜 바꾸는가
10~20분	네 assumption과 primary hypothesis	assumption map	틀리면 어떤 decision이 바뀌는가
20~35분	whole problem·artifact stage	start/end·channel·support	사용자 결과가 끊기지 않는가
35~50분	vertical slice·quality floor	six-layer map·floor register	end-to-end·viable한가
50~65분	scope inventory·priority·exclusion	raw evidence·bucket·revisit	점수 밖 trade-off가 보이는가

시간	행동	남길 evidence	통과 질문
65~75분	dependency·capacity	graph·owner·due·fallback	critical open 0인가
75~85분	test·metric·decision rule	test card·scorecard	결과가 행동을 바꾸는가
85~90분	Gate·version·handoff	pass/fail/TBR·risk	M12-04가 같은 ID로 시작하는가

22.1 실습 중 반드시 남길 여덟 줄

```
Source version:
Product goal + outcome:
Primary hypothesis + disconfirming evidence:
Whole problem start → end:
Vertical slice six layers + quality floors:
Now / next / later / not-now + exclusions:
Test + metric + persevere / pivot / stop:
Gate result + residual risk + M12-04 handoff:
```

23. 제출 전 한 장 점검표

영역	PASS 질문	증거
Trace	M12-02 requirement·constraint·TBR version을 잇는가	Trace matrix
Outcome	User·situation·baseline·target·guardrail이 있는가	Outcome contract
Hypothesis	네 가정과 primary·반증·decision이 있는가	Assumption map
Boundary	Artifact stage와 whole problem start/end가 있는가	Boundary map
Slice	UI·API·data·auth·error·evidence를 관통하는가	Vertical slice
Floor	Access·security·privacy·performance·recovery를 보존하는가	Floor register
Priority	Raw inputs·range·source와 trade-off가 있는가	Evidence sheet
Dependency	Critical open 0, owner·due·fallback이 있는가	Dependency graph
Capacity	Now가 상한 안이고 buffer가 보이는가	Capacity contract
Learning	Smallest test·metric·decision rule이 연결되는가	Scorecard

영역	PASS 질문	증거
Governance	Version·DoD·TBR·risk·authority가 있는가	Decision log
Handoff	PRD·screen·API·data가 같은 ID를 받을 준비가 됐는가	M12-04 map

24. 셀프 테스트 30

답을 보기 전 종이에 한 문장과 합성 사례 ID 하나를 씁니다. 정답을 외우는 대신 내 scope 후보에서 evidence 위치를 표시합니다.

1. MVP를 ‘기능이 가장 적은 첫 버전’이라고만 정의하면 왜 위험한가요?

정답과 해설

최소 기능 수는 학습 질문·사용자 결과·viability를 보장하지 않습니다. MVP는 중요한 가설을 검증할 최소 effort이면서도 사용자가 한 결과를 끝까지 달성하고 품질 하한선을 지킬 수 있어야 합니다.

2. Validated learning이란 무엇인가요?

정답과 해설

관찰 가능한 행동과 결과 evidence로 가정이 맞거나 틀렸음을 확인하고 다음 범위·제품 결정을 실제로 갱신한 학습입니다.

3. Prototype·PoC·pilot·MVP·release를 같은 단계처럼 쓰면 안 되는 이유는 무엇인가요?

정답과 해설

각 artifact는 목적·대상·fidelity·운영·risk·approval이 다릅니다. 앞 단계의 성공은 다음 단계나 공개 release 승인을 자동으로 뜻하지 않습니다.

4. M12-03의 네 assumption 종류를 쓰세요.

정답과 해설

Value, usability, feasibility, viability assumption입니다.

5. Riskiest assumption을 어떻게 고르나요?

정답과 해설

불확실성이 크고 틀렸을 때 user·problem·solution·cost·schedule 결정을 가장 크게 바꾸는 가정을 우선합니다.

6. 좋은 hypothesis의 최소 필드는 무엇인가요?

정답과 해설

대상, 상황, 행동 또는 변화, 예상 outcome, measurement, disconfirming evidence, 결정입니다.

7. Whole problem boundary에 포함할 여섯 요소를 쓰세요.

정답과 해설

Primary user, beneficiary, situation, start, end outcome, channel·support boundary입니다.

8. Horizontal slice와 vertical slice의 차이는 무엇인가요?

정답과 해설

Horizontal slice는 UI나 backend처럼 계층 일부를 넓게 만들고, vertical slice는 UI·API·data·authorization·error·evidence를 관통해 한 사용자 결과를 끝까지 만듭니다.

9. MVP vertical slice에 evidence layer가 필요한 이유는 무엇인가요?

정답과 해설

사용자 결과와 품질을 측정하지 못하면 기능이 작동해도 가설을 검증하거나 다음 결정을 갱신할 수 없기 때문입니다.

10. Quality floor란 무엇인가요?

정답과 해설

기능 우선순위 점수와 교환하지 않고 viable한 사용을 위해 처음부터 지켜야 할 access·security·privacy·performance·reliability·audit·recovery 등의 최소 조건입니다.

11. ‘MVP니까 보안과 접근성은 나중에’가 왜 틀렸나요?

정답과 해설

핵심 flow가 처음부터 일부 사용자를 배제하거나 data·권한 피해를 만들면 viable한 학습이 아니며, 뒤에 붙일수록 구조와 evidence가 왜곡됩니다.

12. Explicit exclusion의 필드는 무엇인가요?

정답과 해설

제외 item, rationale, owner, revisit condition, 영향과 version입니다.

13. Now·next·later·not-now의 역할을 한 문장씩 설명하세요.

정답과 해설

Now는 현재 timebox의 구체적 범위, next는 now evidence 뒤의 가까운 의도, later는 불확실한 먼 가설, not-now는 이유와 revisit가 있는 명시적 제외입니다.

14. MoSCoW의 Must-have test는 무엇인가요?

정답과 해설

해당 item이 없으면 timebox outcome·법적 경계·quality floor·viability가 실제로 실패하는지 묻는 검사입니다.

15. RICE의 네 입력을 쓰세요.

정답과 해설

Reach, impact, confidence, effort입니다.

16. RICE 점수를 decision으로 바로 쓰면 안 되는 이유는 무엇인가요?

정답과 해설

입력 source·범위·불확실성과 risk·dependency·quality floor·strategy·opportunity cost를 한 숫자가 대체할 수 없기 때문입니다.

17. Reach에는 왜 기간과 denominator가 필요한가요?

정답과 해설

대상이 누구 중 몇 명이며 어느 기간의 기회인지 없으면 서로 다른 크기를 같은 숫자처럼 비교하게 됩니다.

18. Effort에 포함할 범위를 쓰세요.

정답과 해설

설계·개발·data·test·accessibility·security·ops·support·documentation과 uncertainty range를 포함한 whole-team effort입니다.

19. Cost of delay와 opportunity cost의 차이는 무엇인가요?

정답과 해설

Cost of delay는 가치가 낮어져 잃는 양이고, opportunity cost는 한 선택 때문에 다른 대안을 하지 못해 포기한 가치입니다.

20. Critical dependency의 최소 필드는 무엇인가요?

정답과 해설

Dependency, predecessor·successor, owner, due date, status, impact, fallback입니다.

21. 합성 candidate의 now effort가 57/100인 이유는 무엇인가요?

정답과 해설

60% 상한 안에서 core flow와 quality floor·instrumentation을 포함하고, 43%를 non-now·buffer·학습·dependency 대응으로 보존하기 위해서입니다.

22. Smallest sufficient test란 무엇인가요?

정답과 해설

기능을 많이 만드는 시험이 아니라 다음 결정을 바꾸기에 충분한 evidence를 얻는 가장 작고 안전한 시험입니다.

23. Learning scorecard의 세 metric 종류를 쓰세요.

정답과 해설

Leading indicator, outcome indicator, guardrail metric입니다.

24. Vanity metric을 피하는 방법은 무엇인가요?

정답과 해설

Metric을 product decision, segment, numerator·denominator, source, cadence, owner, disconfirming evidence와 연결합니다.

25. Persevere·pivot·stop rule은 언제 써야 하나요?

정답과 해설

결과를 보기 전에 threshold·cadence·owner와 함께 선등록해야 편한 evidence만 선택하는 판단을 줄일 수 있습니다.

26. Scope version과 baseline이 필요한 이유는 무엇인가요?

정답과 해설

어떤 포함·제외·가정·effort·evidence 상태를 비교하는지 고정하고 변경 영향과 회귀를 추적하기 위해서입니다.

27. MVP Scope Gate의 네 lane을 쓰세요.

정답과 해설
Outcome·learning, slice·viability, priority·dependency, evidence·governance입니다.

28. 합성 MVP Scope Gate의 핵심 수치 다섯 개를 쓰세요.

정답과 해설
24/24 scenario, critical 100%, now 57/100점, explicit exclusion 4개, unresolved critical dependency 0개입니다. Outcome·slice·floor·priority·test coverage도 100%입니다.

29. `mvp_scope_ready` 가 실제로 뜻하지 않는 것은 무엇인가요?

정답과 해설
실제 MVP, build, public pilot, production release, budget·investment, contract, security·accessibility certification 승인을 뜻하지 않습니다.

30. M12-04에 무엇을 handoff하나요?

정답과 해설
Product goal·hypothesis·scope version·now/next/later/not-now·requirement trace·vertical flow·quality floor·metric·decision rule·dependency·exclusion·TBR를 PRD·screen·API·data 문서와 연결하도록 넘깁니다.

25. 공식 근거와 적용 경계

근거	이 매뉴얼에서 사용한 원리	적용 경계
Lean Startup Co · What is an MVP?	최소 effort로 최대 validated learning	특정 score·Gate를 표준으로 강제하지 않음
Lean Enterprise Institute · Lean Startup	Build·Measure·Learn·validated learning	합성 workflow는 교육용 구현
Agile Alliance · Minimum Viable Product	MVP 용어와 학습 목적	제품·조직 맥락별 적용 필요

근거	이 매뉴얼에서 사용한 원리	적용 경계
GOV.UK · Making prototypes	질문에 맞는 prototype fidelity	Prototype이 MVP·release를 자동 증명하지 않음
GOV.UK · Alpha phase	가장 위험한 가정 시험	Government service stage를 그대로 복제하지 않음
GOV.UK · Deciding on priorities	사용자 필요·risk·dependency로 priority 결정	팀 authority와 policy 필요
GOV.UK · Developing a roadmap	Outcome 중심 roadmap	먼 날짜 확정 약속으로 사용하지 않음
GOV.UK · Planning agile	계획의 반복 갱신	2026-03-31 갱신 페이지 기준
GOV.UK · Solve a whole problem	시작부터 결과·channel·support까지	2026-01-29 갱신 페이지 기준
Scrum Guide 2020	Product Goal·ordered backlog·usable increment·DoD	MVP와 Product Backlog를 같은 것으로 취급하지 않음
Agile Business Consortium · MoSCoW	Must·Should·Could·Won't timebox priority	Must 기준을 팀이 명시해야 함
Intercom · RICE	Reach·impact·confidence·effort 비교	Score는 decision을 대체하지 않음
Product Talk · Opportunity Solution Tree	Outcome·opportunity·solution·assumption test 연결	한 도구를 유일한 discovery 방식으로 강제하지 않음
W3C · WCAG 2.2	Core flow accessibility floor	표준 링크만으로 적합성 claim 불가
OWASP ASVS 5.0.0	검증 가능한 application security floor	인증·penetration test를 대체하지 않음
NIST SSDF SP 800-218	Secure development practice와 evidence	조직 risk profile에 맞춘 tailoring 필요

25.1 이 매뉴얼의 합성 규칙

- 실제 개인정보·참여자 data·meeting 원문·requirement 원문·secret을 사용하지 않습니다.
- 외부 network·production resource·live side effect·자동 연락을 사용하지 않습니다.
- 실제 public pilot·release·contract·budget·investment·security·accessibility·MVP build approval claim을 하지 않습니다.
- RICE·MoSCoW·60% capacity·24 scenario·57 audit는 이 합성 사례의 교육 계약이며 보편 표준 임겟값이 아닙니다.

26. 다음 단계 · M12-04 handoff

M12-04에서는 이 범위 후보를 PRD·화면·API·data 문서로 쪼개되 같은 ID와 version을 유지합니다.

M12-03 product goal + hypothesis + scope version
→ PRD outcome + in/out + metric + decision rule
→ screen flow + UI state + accessibility
→ API request + authorization + error
→ data entity + state + history + retention
→ acceptance + test + evidence + change impact

Handoff item	M12-03 source	M12-04 destination
Product goal·outcome	Outcome contract	PRD overview·metric
Whole problem·vertical slice	Journey·slice map	Screen flow·API sequence
Quality floor	Floor register	Screen·API·data nonfunctional contract
Scope item·exclusion	SCP·EX ID	PRD scope·out-of-scope
Dependency·TBR	Graph·decision log	Document open question·owner
Test·decision rule	Learning scorecard	Acceptance·analytics·review plan

완료 문장: 나는 MVP를 작은 기능 목록이 아니라 outcome·hypothesis·whole problem·vertical slice·quality floor·priority evidence·dependency·capacity·test·decision·version이 연결된 학습 계약으로 설명하고, **mvp_scope_ready** 가 실제 build·pilot·release 승인이 아님을 구분할 수 있습니다.

부록 A. 용어집 300 학습 지도

MVP 범위·우선순위 용어집 300은 15개 묶음으로 구성됩니다. 한 번에 20개만 읽고 반드시 합성 사례의 ID를 붙입니다.

묶음	핵심 질문
01~03	MVP·가설·사용자 결과를 설명할 수 있는가
04~06	Artifact stage·vertical slice·quality floor를 구분하는가

목록	핵심 질문
07~10	Scope·priority·trade-off·dependency를 증거로 쓰는가
11~13	Experiment·metric·decision·roadmap을 연결하는가
14~15	Gate evidence와 M12-04 문서 trace를 설명하는가